

## PAPER: ONDERZOEKSVOORSTEL

# Scripting voor Netwerkbeheer: DNS- en DHCP-Beheer met Python-scripts en IP Address Management Integratie.

Bachelorproef, 2023-2024

Stijn Coppens

E-mail: [stijn.coppens@student.hogent.be](mailto:stijn.coppens@student.hogent.be)

Project repo: <https://github.com/stcoppens/Bachelorproef>

## Samenvatting

Netwerkconfiguraties worden momenteel grotendeels handmatig beheerd, wat inefficiënt is als proces, en tevens gevoelig is voor menselijke fouten. Het minimaliseren van handmatig beheer van netwerkconfiguraties zou leiden tot efficiëntiewinsten, tijdsinsten en verminderde complexiteit. Deze bachelorproef richt zich op het ontwikkelen van een innovatieve, geautomatiseerde benadering voor netwerkbeheer (en IP-adresallocatie), door middel van Python-scripts. Het hoofddoel is het creëren van een abstractielaag boven een bestaande netwerkbeheertool voor DNS en DHCP, waarbij de API van een IP Address Management (IPAM)-tool wordt aangestuurd. Deze abstractielaag maakt interactie met de IPAM-tool mogelijk via Python-scripts en wordt toegepast door API-aanroepen vanuit een intuïtief webportaal.

**Keuzerichting:** System & Network Administrator

**Sleutelwoorden:** DNS, DHCP, IPAM, Python

## Inhoudsopgave

1	Inleiding	1
2	Begrippen	1
2.1	DNS	2
2.2	DHCP	2
2.3	IPAM	2
3	Probleemstelling	2
4	Doelstelling	3
5	Methodologie	3
5.1	Software	3
5.2	Methodiek scripts	3
5.3	Fase 1. Literatuurstudie	3
5.4	Fase 2. Scripts schrijven	3
5.5	Fase 3. Webpagina schrijven	3
5.6	Fase 4. Scriptie	4
5.7	Gantt stappenplan	4
6	Verwachte resultaten	4
7	Discussie, conclusie	4
	Referenties	4

## 1. Inleiding

In de hedendaagse technologische omgeving is netwerkbeheer van cruciaal belang voor de soepele werking van bedrijfsinfrastructuren. Traditioneel vereist het beheren van netwerkconfiguraties, met name *Domain Name System (DNS)* en *Dynamic Host Configuration Protocol (DHCP)*, een grote mate van handmatige inspanning. Dat kan leiden tot inefficiëntie en menselijke fouten. Om dit op te lossen bestaan

er meerdere *Internet Protocol Address Management (IPAM)* softwarepakketten om de taken van netwerkbeheerders te automatiseren.

Dit onderzoek zal een antwoord bieden op de vraag hoe je via python scripts een abstractielaag kan voorzien om via een webapplicatie veelgebruikte acties door te sturen naar de *Application Programming Interface (API)* van een IPAM-tool.

Uit dit project worden volgende voordelen verwacht:

- Verminderde complexiteit voor veelgebruikte acties zoals het reserveren van IP-adressen.
- Tijdswinst door het uitsluiten van handmatig beheer.
- Efficiëntie door het vermijden van menselijke fouten.
- Verbetering van gebruiksvriendelijkheid

Deze voordelen zullen bijdragen aan een gestroomlijnde netwerkinfrastructuur.

## 2. Begrippen

Internet Protocol (IP) is het fundament van elk gestructureerd, goed functionerend en veilig netwerk. Het geeft de mogelijkheid efficiënt gegevens te routeren, netwerken te verdelen in meer beheersbare eenheden, toegang te beperken tot

gevoelige data of systemen, services te identificeren en het oplossen van netwerkproblemen (Postel, 1981). Dit hoofdstuk legt uit wat DNS en DHCP is en waarom IPAM helpt bij het beheren van IP netwerken.

## 2.1. DNS

Mockapetris (1987) schrijft dat DNS een systeem is dat *resource records* gebruikt om onder andere vertalingen te voorzien tussen domeinnamen en IP-adressen. Als voorbeeld kan je via de browser naar google surfen via het IP-adres 142.251.36.35 of via het domeinnaam *www.google.be*.

Zoals beschreven door Mockapetris (1987) voorziet DNS meerdere types resource records die netwerkbeheerders kunnen meegeven:

- **A:** Dit resource record beschrijft een host adres. Vb. "*server1.voorbeeld.com. IN A 192.168.1.1*" maakt de vertaling zodat het toestel met de domeinnaam *server1.voorbeeld.com* bereikbaar is zowel via het IP-adres 192.168.1.1 als via de domeinnaam.
- **CNAME:** Dit resource record beschrijft de kanonieke naam van een host, het wordt gebruikt om een alias of subdomein naar het hoofddomein door te verwijzen. Vb. "*www.voorbeeld.com. IN CNAME server1.voorbeeld.com*" zorgt dat *server1* ook bereikbaar is via "*www.voorbeeld.com*".
- **MX:** Dit resource record is een *mail exchange* record en wordt gebruikt om aan te geven welke mailservers verantwoordelijk zijn voor het ontvangen van mails binnen een domein. Vb. "*voorbeeld.com. IN MX 10 mailserver.voorbeeld.com*" geeft de DNS server mee welke server de mailserver is.
- **NS:** Dit resource record is een *name server* record, het beschrijft welke DNS-servers verantwoordelijk zijn voor het beheren van DNS-informatie voor een domein. Vb. "*voorbeeld.com. IN NS dns1.voorbeeld.com*" verwijst naar *dns1* als DNS-server voor het domein "*voorbeeld.com*".
- **PTR:** Dit resource record is een *Pointer* record, het wordt gebruikt om via IP een vertaling te vragen aan de DNS-server in plaats van via de naam.
- **SOA:** Dit resource record is een *Start of Authority* record die belangrijke informatie bevat over de zone, zoals welke de primaire DNS-server, contactpersonen, etc. zijn.

## 2.2. DHCP

Dit protocol voorziet een framework voor het doorgeven van configuratie informatie naar

hosts (lees: computers) op het netwerk. Zo kan een computer bijvoorbeeld een IP-adres ontvangen waarmee die kan communiceren binnen het netwerk waarop die is aangesloten (Droms, 1997).

IP-netwerken worden door netwerkbeheerders op een logische manier opgesplitst in subnetwerken. Hierbij worden de beschikbare IP-adressen verdeeld in subnetwerken (subnet). Toestellen binnen subnet A zullen elkaar kunnen bereiken terwijl een toestel in een subnet B zonder de nodige routing geen verbinding zal kunnen maken met de toestellen in subnet A.

Voor DHCP zullen netwerkbeheerders subnets (of pools van IP-adressen) aanbieden aan de DHCP-server. Die zal gebruik maken van deze pools door (onder andere) IP-adressen uit te delen aan toestellen die verbinden op het netwerk en daarbij de DHCP-server laten weten dat ze nog geen IP-adres hebben.

Droms (1997) schrijft dat DHCP drie mechanismes gebruikt voor het uitdelen van IP-adressen:

- **Automatisch:** Permanent toewijzen van een IP-adres.
- **Dynamisch:** IP-adres voor een bepaalde tijd toewijzen.
- **Manueel:** Een (door de netwerkbeheerder) vooraf bepaald IP-adres toewijzen, in vakjargon noemt met dit een IP-reservatie.

## 2.3. IPAM

Naast de vele uitdagingen die zowel DNS als DHCP met zich meebrengen, is het beheren van de vele DNS records, IP-adres ranges en de vaak vele IP-reservaties zeker iets waar een netwerkbeheerder over moet waken. Een mogelijke oplossing hiervoor is het gebruiken van IP Address Management (IPAM) via softwarepakketten die IPAM aanbieden. IPAM laat toe IP-adressen efficiënt te beheren in een netwerk, het leidt tot een gestructureerde aanpak waardoor conflicten tussen subnetten worden vermeden. Het geeft een compleet overzicht van het netwerk met percentages van hoeveel adressen beschikbaar en in gebruik zijn. IPAM geeft eveneens de mogelijkheid om de historie bij te houden waardoor het van pas komt voor schaalbaarheid en beveiliging van het netwerk (Rooney & Dooley, 2020).

## 3. Probleemstelling

Deze bachelorproef zal uitgevoerd worden bij Universiteit Gent (UGent), directie ICT. Momenteel werkt UGent met scripts die op basis van zo genaamde subnetbestanden de nodige acties uitvoeren op de DHCP en DNS servers.

Deze subnetbestanden stellen elk een subnet voor en beschrijven cruciale informatie zoals DNS-servers, welk *Virtual Local Area Network*

(VLAN) nummer, gateway, etc. Hiernaast bevatten deze zowel alle beschikbare als gereserveerde IP-adressen met daarbij eventueel enkele regels voor DNS en beveiliging.

Voor elke IP reservatie die moet gebeuren, krijgt het netwerkteam via een webportaal van intern UGent-personeel een mail met daarin de nodige hostinformatie die ze in het daarvoor bestemde subnetbestand plakken. Indien hier nog extra DNS- of beveiligingsregels bij horen, moet de netwerkbeheerder deze er zelf nog bij schrijven.

Het overzicht van de beschikbare IP ranges is beschreven in een interne wikipediapagina met daarbij de beschrijving van elke range. Dit brengt meerdere uitdagingen met zich mee:

- **Tijd:** Het onderhouden van de scripts, subnetbestanden, IP reservaties (maken en opkuisen) kan veel tijd vragen.
- **Schaalbaarheid:** Doordat elke wijziging het bestaande bestand wijzigt en er dus geen historische data is kan men moeilijk trends herkennen, ook wikipedia pagina's moet men manueel bijwerken bij grote wijzigingen in de structuur.
- **Consistentie:** De huidige aanpak vraagt meerdere manuele acties, wat deze aanpak vatbaar maakt voor menselijke fouten of vergissingen.
- **Beveiliging:** Het bewaren van IP-gegevens in ongecrypteerde bestanden kan leiden tot misbruik.

UGent is momenteel stappen aan het ondernemen voor het implementeren van *Efficiënt IP (EIP)*, een IPAM-softwarepakket, in hun opzet. Dankzij deze implementatie is de verwachting dat de hierboven beschreven indicatoren zullen verbeteren.

## 4. Doelstelling

Deze bachelorproef zal een abstractielaag maken boven EIP waarbij python scripts via de *Application Programming Interface (API)* van EIP commando's zullen uitvoeren op EIP. Door de omvang van het EIP-project is het binnen de voorziene tijd van de bachelorproef niet haalbaar om UGent volledig over te zetten op de werking van EIP. Aangezien dit kritische componenten zijn, zal alles eerst uitvoerig getest worden waarbij elke stap in de migratie naar EIP weloverwogen is. Daarom stel ik als doel om een eerste versie op te leveren van een webportaal waarop men reeds meerdere IP-reservaties kan aanmaken, wijzigen of verwijderen. Netwerkbeheerders zullen op deze eerste versie de mogelijkheid hebben om een overzicht te krijgen van alle openstaande aanvragen en deze al dan niet goed te keuren.

## 5. Methodologie

In dit hoofdstuk wordt beschreven welke software gebruikt wordt, hoe deze wordt toegepast, en welke fases het project zal doorlopen.

### 5.1. Software

Alle scripts worden geschreven in **Visual studio code** in de programmeertaal **Python**. Voor de testen om rechtstreeks op maat gemaakte commando's te sturen wordt er gebruik gemaakt van **Postman**.

### 5.2. Methodiek scripts

Voordat een script wordt geschreven, worden eerst gerichte testen gedaan waarbij commando's met parameters naar de API van EIP worden verzonden. Hierbij wordt er zowel naar de resultaten van de API gekeken als naar wat er op EIP zelf gebeurt via de webpagina. Eens deze testen voor een commando afgelopen zijn, wordt er pas overgegaan tot het schrijven van het python script. Hierbij worden telkens de parameters binnen elk script opgezet volgens de voorwaarden van de API.

### 5.3. Fase 1. Literatuurstudie

Voor deze eerste fase worden veertien dagen uitgerokken. In deze fase wordt gezocht naar documentatie en literatuur van gelijkaardige projecten waarbij een webpagina met formulieren en logins gebruiken om scripts aan te roepen. Daarnaast wordt er ook uitgebreid aandacht gegeven aan de literatuur van Efficiënt IP zelf om na te gaan welke eisen deze stelt voor het aanmaken, wijzigen en verwijderen van IP-adressen. Op het einde van deze fase zal er een verslag geschreven zijn met alle belangrijke punten die zijn meegenomen uit de literatuur.

### 5.4. Fase 2. Scripts schrijven

Binnen de tweede fase wordt het grootste stuk van de python scripts geschreven. Deze fase voorziet telkens één week om een script te schrijven en daarop aansluitend één week om het geschreven script te testen en te troubleshooten. De drie scripts zijn:

- Maken van een IP reservatie
- Wijzigen van een bestaande IP reservatie
- Verwijderen van een bestaande IP reservatie

### 5.5. Fase 3. Webpagina schrijven

Het schrijven van de webpagina begint halverwege fase twee aangezien deze nauw op elkaar aansluiten. Hiervoor worden er drie weken voorzien voor het schrijven vanop de hoofdpagina van waar de drie scripts kunnen worden aangeroepen. Hierna zijn er nog drie weken voorzien

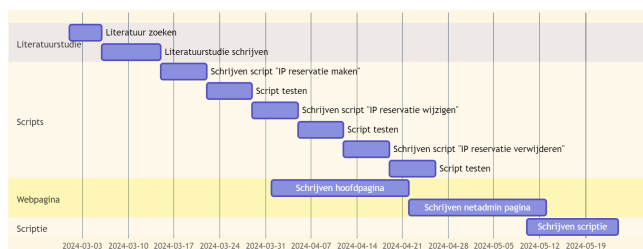
voor het schrijven van de netwerkbeheerder pagina waarop die de goedkeuring moet geven voor de gevraagde wijzigingen en eventueel zelf nog aanpassingen kan doen.

## 5.6. Fase 4. Scriptie

De laatste dagen van het project worden gevuld met het bundelen van alle notities uit voorgaande fases en het schrijven van de scriptie.

## 5.7. Gantt stappenplan

Een overzicht van alle fases worden weergegeven in dit Gantt stappenplan



Figuur 1: Gantt Stappenplan

## 6. Verwachte resultaten

De verwachte resultaten omvatten een succesvolle implementatie van de abstractielaag boven de bestaande netwerkbeheer tool, waardoor interacties met EIP mogelijk zijn via Python-scripts. Het intuïtieve webportaal zou een verbeterde gebruikerservaring moeten bieden door middel van geoptimaliseerde API-aanroepen. Verder zou de automatisering van netwerkconfiguraties, met name IP-adresallocatie, moeten leiden tot verminderde complexiteit, verbeterde efficiëntie en algemene gebruiksvriendelijkheid in het netwerkbeheerproces.

## 7. Discussie, conclusie

Dankzij het implementeren van deze webpagina met de onderliggende kunnen medewerkers van Universiteit Gent eenvoudig, consistent, en snel IP reservaties maken, wijzigen en verwijderen. De netwerkbeheerders kunnen al deze wijzigingen dan op hun eigen webpagina controleren, aanpassen en beoordelen. Na dit onderzoek zullen er nog voldoende mogelijkheden zijn om de webpagina aan te vullen met extra functies tot de mate dat de netwerkbeheerders zelf geen wijzigingen meer moeten doen binnen de IPAM tool zelf.

## Referenties

Droms, R. (1997). *Dynamic Host Configuration Protocol* (tech. rap.). RFC Editor. <https://doi.org/10.17487/RFC2131>

Mockapetris, P. (1987). *Domain names-concepts and facilities* (tech. rap.). <https://doi.org/10.17487/RFC1034>

Postel, J. (1981). *Internet protocol* (tech. rap.). <https://doi.org/10.17487/RFC0791>

Rooney, T., & Dooley, M. (2020). *IP Address Management*. John Wiley & Sons. <https://doi.org/10.1002/9781119692263>