

System Engineering Project.

Projectbrochure studiewijzer.

Professionele bachelor toegepaste informatica

Academiejaar: 2022–2023

Departement IT en Digitale Innovatie.

**HO
GENT**

Inhoudsopgave

1	5
VOORWOORD.....	5
2	7
STUDIEWIJZER	7
2.1. DOEL EN PLAATS VAN HET OPLEIDINGSONDERDEEL	7
2.2. LEERDOELEN EN COMPETENTIES	7
2.3. REGLEMENTEN VAN TOEPASSING	9
2.4. LEERMATERIAAL	9
2.5. WERK- EN LEERAANWIJZINGEN	10
2.6.1. COMMUNICATIE EN BEDRIJFSCULTUUR	11
2.7. STUDIEBEGELEIDING EN PLANNING	12
2.7.1. GROEPEN.....	12
2.7.2. AANWEZIGHEID.....	13
2.7.3. PLANNING.....	14
2.8. EVALUATIE.....	16
2.8.1. EERSTE EXAMENKANS.....	16
2.8.2. TWEEDE EXAMENKANS	17
2.8.3. INHAALKANSEN	18
2.8.4. DOSSIER	18
3	21
OPGAVE.....	21
3.1. ALGEMEEN.....	22
3.2. NETWERK	22
3.3. DOMAIN CONTROLLER.....	24
3.4. WEBSERVER	25
3.5. EMAIL.....	26
3.6. MATRIX.ORG SERVER	26
3.7. EIGEN KEUZE	26
3.8. CISCO LABO'S	27
3.8.1 LABO 1	27
3.8.2 LABO 2	33
4	36
TOOLS, TEAMWORK	36

4.1. PROFESSIONEEL GEBRUIK VAN GIT.....	36
4.1.1. VOORBEREIDING	36
4.1.2. ALGEMENE RICHTLIJNEN	37
4.1.3. BRANCHING & MERGING, PULL REQUESTS	38
4.1.4. TRUNK BASED DEVELOPMENT	40
4.2. KANBAN	42

1

Voorwoord

Beste student(e),

Dit document bevat de richtlijnen voor het opleidingsonderdeel *System Engineering Project*, zowel voor de **reguliere** studenten als studenten via het traject van **afstandsleren**. De organisatie verloopt grotendeels gelijkaardig, en als er verschillen zijn, worden die duidelijk vermeld in deze studiewijzer.

Dit project heeft als belangrijkste doelstelling het in team opzetten van **complexe en geïntegreerde ICT-infrastructuur** die beschikbaar is voor eindgebruikers. We leggen daarbij bijzondere nadruk op:

- Het **integreren** van verschillende netwerkdiensten tot één werkend geheel;
- Verregaande **automatisering** zodat het in productie brengen van ICT-infrastructuur voorspelbaar en reproduceerbaar wordt;
- Goede **documentatie**, testplannen en -rapporten;
- Een systematische en grondige aanpak bij **troubleshooting**;
- Als individu functioneren in een groter **team**;
- Omgaan met werken op afstand en een goede samenwerking en communicatie bewaren.

Je maakt deel uit van een projectgroep van maximaal 5 leden. Je bent minstens een halve dag per week, tenzij anders gecommuniceerd, met jouw groep aanwezig op de campus om aan het project te werken. Dit geldt niet voor de studenten via afstandsleren. Elke projectgroep wordt toegewezen aan een begeleidende lector die jullie technisch zal opvolgen. Ook het proces wordt opgevolgd door een lector analyse.

Je moet te allen tijde kunnen aantonen dat je een niet-triviale, zinvolle bijdrage geleverd hebt aan het project. Dat gebeurt niet alleen aan de hand van de code die je geschreven hebt, maar ook je technische documentatie en testplannen. Tenslotte moet jouw deel van het werk ook geïntegreerd zijn met de rest van het uit te werken project. Jouw bijdrage is dus nodig voor het slagen van het project als geheel!

Bij de eindbeoordeling houden we niet alleen rekening met het eindproduct, maar ook met het gevolgde proces, de individuele inbreng, de houding tijdens het verloop van de projectwerking en de vorderingen gemaakt door de individuele student.

Misschien heb je na het lezen van deze brochure toch nog vragen. Een deel daarvan zal sowieso tijdens de loop van het project duidelijk worden. Verder kan je altijd vragen stellen aan je begeleiders, of bij de coördinator van het project, Sebastiaan Labijn (sebastiaan.labijn@hogent.be).

Veel succes!

De begeleidende lectoren System Engineering Project:

- Technische begeleiding:
 - Jeroen Courtens
 - Thomas Parmentier
 - Olivier Rosseel
 - Martijn Saelens
 - Karine Van Driessche
- Begeleiding Analyse:
 - Guy Dekoning
 - Sebastiaan Labijn

2

Studiewijzer

2.1. Doel en plaats van het opleidingsonderdeel

Dit opleidingsonderdeel behoort tot de leerlijn systeem- en netwerkbeheer binnen de bacheloropleiding toegepaste informatica. In het systeembeheerproject toon je aan dat je bekwaam bent om aan de slag te gaan als een (junior) systeem- of netwerkbeheerder, system engineer, of gelijkaardige profielen.

Binnen het project zal je de competenties die je in andere opleidingsonderdelen verworven hebt verder uitdiepen en uitbreiden. Daarnaast wordt er ook bijzondere aandacht geschonken aan het *proces*, bv. functioneren in een team, rapporteren naar leidinggevend en planning.

2.2. Leerdoelen en competenties

Zie ook de studiefiche¹ of de Chamilo-cursus.

- Technische competenties:
 - Kan zelfstandig complexe fysieke netwerken opzetten en configureren
 - Kan zelfstandig complexe fysieke ICT-infrastructuur (netwerk + services) opzetten, beveiligen en configureren
 - Kan verschillende componenten waaruit ICT-infrastructuur bestaat (netwerk + services) met elkaar integreren zodat ze als geheel beschikbaar zijn voor de eindgebruikers

¹ Dat kan via de Chamilo-cursus of <https://www.hogent.be/studiefiches/>

- Kan problemen in complexe ICT-infrastructuur (netwerk + services) opsporen en oplossen door het toepassen van een systematische en grondige aanpak
- Kan de meest geschikte documentatie voor een specifieke situatie (meer bepaald de Linux-distributie, softwareversies, enz.) opzoeken en gebruiken.
- Kan typische services zoals DNS, DHCP, Fileserver, ... opzetten, configureren en beschikbaar maken over een netwerk
- Kan de uitgewerkte configuratie adequaat documenteren zodat deze reproduceerbaar is door een collega
- Kan een testplan opstellen binnen het expertisegebied voor een eenvoudig project
- Kan een testplan opstellen dat aantoonst in hoeverre de opgezette systemen voldoen aan de specificaties
- Kan een versiecontrole-tool gebruiken om efficiënt in team te werken
- Projectmanagement en professioneel handelen:
 - Kan werken volgens de principes van agile strategie binnen het expertisegebied (KanBan) in een ICT-project
 - Kan spontaan en actief gericht op zoek gaan naar een creatieve oplossing
 - Kan een korte tot middellange termijnplanning maken voor een eenvoudige opdracht onder begeleiding van de onderwijsgevende
 - Kan feedback aanvaarden en verwerken en spontaan opbouwende feedback geven
 - Kan zelfstandig werken volgens planning op middellange termijn en afspraken, kan zich spontaan voorbereiden op overleg, kan een retrospectieve organiseren en plus- en pijnpunten van de organisatie duiden
 - Kan spontaan professioneel handelen in elke situatie naar medestudenten, onderwijsgevenden en klanten
 - Kan spontaan reflecteren onder begeleiding van een onderwijsgevende
 - Kan zelfstandig de kwaliteit van het eigen werk opvolgen en past spontaan het werk aan de eisen van de klant
 - Kan stimulerend samenwerken met anderen en actief bijdragen aan teamwork. Het team is zelfsturend onder begeleiding van de onderwijsgevende.

2.3. Reglementen van toepassing

Uiteraard is het gehele Onderwijs- en Examenreglement ook van toepassing op dit opleidingsonderdeel, maar we willen graag expliciet de aandacht vestigen op de hieronder opgesomde reglementen. **Lees deze goed na!** Wij veronderstellen dat jullie de regels kennen en ook de consequenties van het niet naleven ervan.

- Onderwijs- en Examenreglement (OER):
 - Art. 33, *Afwezigheid bij evaluatie*;
 - Art. 36, *Vermelding 'afwezig'*;
- Departementaal Reglement (DOER) van departement IT en Digitale innovatie:
 - Art. 4, *Deelname aan onderwijs- en evaluatieactiviteiten*
 - Art. 6, *Gevolgen bij afwezigheid bij evaluatie*

De reglementen zijn te vinden via de HOGENT-website¹. Verder is het ook belangrijk om de inhoud van de studiefiche te kennen².

2.4. Leermateriaal

Voor dit opleidingsonderdeel is er -behalve deze studiewijzer- geen apart leermateriaal voorzien. Gebruik waar nodig de leermaterialen van de gerelateerde opleidingsonderdelen: Computer Networks, Operating Systems, Linux, Windows Server, Functional Analysis, enz.

Verder is het de bedoeling dat je de informatie die nodig is om de opdracht uit te voeren zelfstandig gaat opzoeken via de bronnen die in de beroepspraktijk gebruikt worden: vakliteratuur, het Internet, enz.

Voor het opzetten van ICT-infrastructuur in de cloud kan je gebruik maken van Infrastructure-as-a-Service platformen als AWS, Azure, Google Cloud, Digital Ocean, enz. Elk van deze providers heeft een gratis versie voor studenten.

¹ <https://www.hogent.be/student/een-vlotte-start/onderwijs-en-examenregeling/>

² Dat kan via de Chamilo-cursus of <https://www.hogent.be/studiefiches/>

2.5. Werk- en leeraanwijzingen

Let erop dat de opgave dit jaar een stuk uitdagender is dan vorig jaar. Waar tijdens System Engineering Lab elke deelopdracht min of meer onafhankelijk van elkaar uit te werken was, moeten de individuele componenten van deze opdracht (netwerkapparatuur en -services) als geheel samenwerken. Het integreren van deze componenten zorgt voor extra complexiteit en maakt ook dat je van elkaars werk afhankelijk bent om het geheel tot een goed einde te brengen.

Werk dus regelmatig aan het project en plan ook vaste momenten in wanneer je aan het project werkt. Spreek ook onderling binnen het team af om regelmatig te overleggen en samen te werken. Je kan op de campus of bv. via Microsoft Teams overlegmomenten inplannen.

De begeleiders rekenen erop dat je intussen voldoende maturiteit hebt om je ook nu al, tijdens het project, op een professionele manier te gedragen.

Daaronder verstaan we onder andere:

- Dat je je respectvol gedraagt ten opzichte van teamleden en begeleiders;
- Dat je zorg draagt voor het dure materiaal dat je ter beschikking krijgt voor het project;
- Een goede werkethiek: presteer de nodige uren en lever een tastbare bijdrage aan het teamwerk;
- Het opnemen van je verantwoordelijkheid voor jouw taken;
- Dat je initiatief neemt i.p.v. te wachten tot iemand je zegt wat je moet doen;
- Dat je niet op je lauweren gaat rusten als je deelopdracht “lichter” blijkt dan die van andere teamleden, maar dat je proactief bespreek met het team of de begeleiders wat je nog meer kan doen;
- Dat je je bijdrage regelmatig commit op de team-repository. Niet enkel tijdens de contactmomenten, maar *telkens je aan het project werkt*. Als je bijvoorbeeld niet aanwezig kan zijn op een contactmoment, kunnen je teamleden toch gebruik maken van jouw werk;
- Open communicatie: bijvoorbeeld, als je afwezig bent op een contactmoment, dan laat je dit weten aan je teamleden zodat zij dit kunnen opvangen; of, als je de afgelopen week weinig of geen resultaten behaald hebt, dan kom je daar voor uit en vraag je zo nodig hulp.

2.6.1. Communicatie en bedrijfscultuur

Alle technologische vooruitgang ten spijt, blijft de belangrijkste uitdaging bij de uitvoering van grote projecten nog altijd communicatie en bedrijfscultuur.

In dit project werken we samen in grotere en verspreide teams. Een goede, open communicatie is daarbij essentieel. Het is de verantwoordelijkheid van ons allen om er voor te zorgen dat niemand reden heeft om zich uitgesloten of onvoldoende geïnformeerd te voelen.

Enkele richtlijnen:

- Gebruik **geen Facebook** of andere sociale media voor onderlinge communicatie, maar enkel kanalen die geschikt zijn voor professioneel gebruik (zie verder).
- De begeleiders moeten toegang hebben tot *alle* communicatiekanalen die het team gebruikt. Het team is verantwoordelijk voor het verlenen van deze toegang en verzorgt dit op eigen initiatief.
- Als je mondeling afspraken maakt met teamleden of begeleiders, maak dan onmiddellijk een verslag dat gepubliceerd wordt voor heel het team. In het bijzonder: van de wekelijkse intervisiegesprekken wordt er telkens een verslag gemaakt door één van de teamleden en zo snel mogelijk gepubliceerd op de team-repo.
- Gebruik zoveel mogelijk elektronische vormen van communicatie, zodat teamleden op een andere locatie de kans hebben om dit mee te volgen, of op zijn minst toegang hebben tot de neerslag ervan.

Onder “bedrijfscultuur” verstaan we de manier waarop mensen binnen een organisatie (werknemers, leidinggevenden, klanten, ...) met elkaar omgaan. Dit is bepalend voor de sfeer en het welbevinden van alle teamleden. Wij willen er naar streven om voor een aangename werksfeer te zorgen waarin we elkaar motiveren om te excelleren.

- We behandelen elkaar met respect, ook al zijn we het niet met elkaar eens.
- Personen worden niet met de vinger gewezen als er iets fout gaat. Fouten of problemen zien we als een opportuniteit om ervaring op te doen en van elkaar te leren. We steken geen energie in het afschuiven van schuld of verantwoordelijkheid, maar zoeken naar manieren om dit in de toekomst te vermijden.
- Geef elkaar ook de ruimte om een bijdrage te leveren aan het geheel. Teamleden die technisch sterker staan mogen de anderen gerust helpen, maar moeten het niet overnemen. Wees een mentor voor je teamgenoten, geen pletwals...

2.7. Studiebegeleiding en planning

Tijdens het semester worden jullie begeleid door de lectoren van de andere opleidingsonderdelen binnen het keuzepakket (Computer Networks, Linux en Windows Server) en leerlijn Grow. Zij nemen de rol aan van *opdrachtgevers* en geven inhoudelijke, technische begeleiding en coachen jullie eveneens in teamwerk en projectmanagement.

Wekelijks is er een intervisiegesprek waar de teams hun vooruitgang bespreken, zowel op vlak van het product als het proces. Het team neemt het initiatief, bepaalt de agenda en is verantwoordelijk voor de notulen. Alle teamleden stellen hun individuele bijdrage van de afgelopen periode voor, welke resultaten geboekt werden, welke problemen ze tegenkwamen, enz. Het team stelt ook doelen voor de komende sprint en verdeelt de taken evenredig.

De begeleiders adviseren en kunnen door de studenten geconsulteerd worden bij problemen. Dit is een verschil met het vorige project, waar de begeleiders het gesprek grotendeels leidden.

De intervisiegesprekken vinden plaats via videoconferenties (Microsoft Teams) of op campus, afhankelijk van de richtlijnen en afspraken met je begeleiders, op een vast moment tijdens de geroosterde contactmomenten. Het precieze tijdstip wordt afgesproken met de begeleiders van je team.

De begeleiders observeren in eerste instantie en pikken in waar nodig of op vraag van de studenten. Tijdens de intervisiegesprekken bespreekt het team wat er de afgelopen week is gerealiseerd en maakt afspraken voor de komende periode. Reflecteer hier ook over. Wat ging moeilijk? Wat hebben jullie geleerd uit de opgedane ervaring? Waar zijn nog problemen mee en hoe ga je die aanpakken? Kan de werking van het team nog verbeterd worden? Enz.

Als je een technische vraag wil stellen buiten de contacturen, of over een onderwerp buiten de expertise van de begeleider van je groep, dan kan je gebruik maken van de chat-kanalen van het algemene Microsoft Team voor dit opleidingsonderdeel (dus liever niet in het Team van jouw projectgroep).

2.7.1. Groepen

Een projectgroep bestaat uit 5 studenten. Voor de kick-off kan je je via Chamilo inschrijven in één van de voorziene groepen. Studenten die in de tweede week nog niet toegewezen zijn aan een groep, worden door de begeleiders ingedeeld. We streven er daarbij naar om elke groep “op te vullens”. Kleinere groepen

kunnen enkel als er nog onvoldoende studenten ingeschreven zijn om aan te sluiten.

Studenten die bij aanvang van het semester nog niet ingeschreven zijn en willen aansluiten bij een bepaalde groep, kunnen dat laten weten aan de coördinator van het project, [Sebastiaan Labijn](mailto:sebastiaan.labijn@hogent.be) (sebastiaan.labijn@hogent.be). Stuur een mail met de andere groepsleden in cc: en vermeld zeker ook de naam van de groep.

Als je de cursus in **afstandsleren** volgt, is de kans reëel dat er geen volledige groepen gevormd kunnen worden. In dat geval wordt de scope van de opgave in overleg met de begeleiders aangepast.

2.7.2. Aanwezigheid

De beoordeling van het opleidingsonderdeel gebeurt op basis van “permanente evaluatie”. Dit betekent dus dat je tijdens het semester voortdurend geëvalueerd zal worden op basis van observatie, enerzijds tijdens de contactmomenten (ook als die on-line verlopen), maar anderzijds ook via de tastbare bijdragen die je levert aan het project (GitHub commits, documentatie, ...).

De wekelijkse intervisiegesprekken gaan, als de voorschriften dit toelaten, steeds op campus door voor de reguliere studenten. Voor studenten afstandsleren gaan deze on-line door. Er zijn echter enkele momenten tijdens het semester voorzien voor een contactmoment op de campus. Tijdens deze on-campus contactmomenten krijgen jullie de kans om met fysieke netwerkkapparatuur te werken om de opgegeven labotaken uit te voeren.

Als je niet aanwezig kan zijn op een afspraak, moet je dat ook altijd wettigen (OER, art. 33), ook al vinden de opvolgingsgesprekken niet plaats op de campus. Indien je onmogelijk een bepaalde afspraak kan nakomen, verwittig dan tijdig je teamleden en je begeleider(s).

We verwachten dat jullie ook buiten de contacturen verder werken aan jullie opdrachten. Hou een gedetailleerd weekrapport bij, zoals we dat ook in System Engineering Lab gevraagd hebben. Op het einde van het semester zouden je gerapporteerde uren (incl. de duurtijd van de kick-off en alle contactmomenten) de totale studietijd voor het vak moeten benaderen, nl. 200 uren. Als je uren rapporteert, dan is er ook een tastbare neerslag van het geleverde werk. Dat kan code zijn, documentatie, testplannen of -rapporten, enz. Ook van onderzoekswerk om informatie te verzamelen over een opdracht die je moet

uitvoeren maak je een verslag (met bronvermeldingen!) zodat jij én je teamleden gebruik kunnen maken van de kennis die je opgedaan hebt.

Ongewettigde afwezigheid op de contactmomenten en/of een onvoldoende of onbewezen bijdrage aan het project kan als gevolg hebben dat je als examencijfer de vermelding “Afwezig” krijgt. Lees in dit verband ook het Departementaal Onderwijs- en ExamenReglement (DOER) na, in het bijzonder art. 4, § 3 over betrokkenheid bij groepswork.

2.7.3. Planning

Week 1: kick-off

- Toelichting organisatie door de begeleiders
- Toelichting opgave
- Inschrijven in groep

Vóór contactmoment Week 2

- Verdelen in groepen + toewijzen taken
- Opzetten werkomgeving
 - GitHub: team repo
 - KanBanbord (Jira, gekoppeld aan team repo)
 - Persoonlijke timesheet (via tijdsregistratie in Jira)
- Starten met werken aan taken: info verzamelen, ...
- Weekverslag aanvullen
 - timesheet
 - gerealiseerde taken
 - vragen voor begeleiders (over opgave, technische moeilijkheden, inhoud studiewijzer, organisatie, ...)
 - planning voor volgende periode (= een week)

Week 2

- Eerste intervisiegesprek:
 - Heeft iedereen voldoende informatie over requirements?
 - Stel zo nodig vragen aan de begeleiders over onduidelijkheden in de opgave

- Maak afspraken over het gebruik van de “project management tools”: Microsoft Teams, GitHub issues, KanBan-bord, Git (branches?, PRs?, trunk based?)
- Technische documentatie opstarten: overzicht netwerk, IP-adrestabel, afbeelding netwerkstructuur, algemene structuur, ...

Week 3 t/m 12: Tussentijdse sprints

Tussen contactmomenten door:

- individueel werken aan eigen taken;
- logboek en weekverslag consequent bijhouden (realisaties, tijdregistratie, planning, vragen);
- Tijdens contactmomenten:
 - team stand-ups
 - intervisiegesprekken met begeleiders (a.d.h.v. logboek);
 - help elkaar met technische problemen;
 - maak afspraken met elkaar waar nodig, zowel op technisch vlak als organisatorisch. Elke student wordt aangemoedigd om verbeteringen voor te stellen in de organisatie van het team. Bespreek dit onderling en/of met de begeleiders;
 - neem nieuwe taken op indien alles afgewerkt is (in samenspraak met het team en/of begeleiders);
 - **begeleiders kunnen ten allen tijde nieuwe taken en/of gewenste functionaliteiten toevoegen aan de opdracht.**

On-campus contactmomenten

Voor zover de covid-maatregelen dit toelaten, verwachten we jouw aanwezigheid minstens een halve dag per week op de campus (voor TIAO volgens afspraak met de begeleiders). Die momenten zijn er om in team aan het project te werken waarbij ook de (netwerk)apparatuur ingezet kan worden. We maken van de gelegenheid gebruik om ook de intervisiegesprekken dan in te plannen.

We koppelen hier ook een retrospectieve aan over het verloop van het project tot nu toe en zullen jullie vragen jezelf en je team te evalueren aan de hand van de rubrics die gehanteerd worden bij de evaluatie.

De exacte planning van deze contactmomenten wordt via Chamilo gecommuniceerd.

Week 12 of 13: finale oplevering

Op het einde van het semester demonstreren jullie het. Toon aan dat jullie het gehele netwerk volledig geautomatiseerd en “from scratch” kunnen opzetten. De tijd is beperkt dus bereid dit goed voor.

Voor de reguliere studenten zal dit normaal doorgaan in week 13, planning volgt via Chamilo. Voor studenten afstandsleren wordt er in onderling overleg een moment afgesproken.

2.8. Evaluatie

Dit opleidingsonderdeel wordt volledig beoordeeld op basis van permanente evaluatie, zowel in de eerste als de tweede examenkans.

2.8.1. Eerste examenkans

De begeleiders beoordelen in onderling overleg enerzijds de teams als geheel en anderzijds de individuele teamleden. Ze gebruiken hiervoor een evaluatiekaart gebaseerd op zgn. *rubrics*. Rubrics zijn een evaluatiesysteem waar je volgens verschillende categorieën en aan de hand van duidelijk beschreven indicatoren beoordeeld wordt op de mate waarin je de beoogde leerresultaten verworven hebt. De begeleiders beoordelen voor elke categorie welk niveau je behaald hebt. Voor enkele beoordelingscriteria moet elke student *individueel* kunnen aantonen dat zij of hij aan de verwachtingen voldoet om te kunnen slagen. Dewelke dit zijn is aangegeven in de evaluatiekaart. Manieren om de eigen bijdrage aan te tonen zijn bijvoorbeeld de opgeleverde code en documentatie (na te gaan via historiek van het versiebeheersysteem), verslagen van de wekelijkse opvolgingsgesprekken, je weekverslagen of observatie door de begeleiders tijdens de evaluatiemomenten.

Aan de hand van een rekenregel worden de beoordelingen voor alle criteria omgezet in een examencijfer. De evaluatiekaart met alle criteria en indicatoren, en rekenregels om het examencijfer te bepalen zal worden gepubliceerd op Chamilo.

Studenten die onvoldoende betrokkenheid (DOER, art. 4, §3) vertonen bij het projectwerk, zullen uitgenodigd worden tot een functioneringsgesprek. De lectoren kunnen hier het initiatief toe nemen, maar studenten kunnen ook melden als een teamlid onvoldoende bijdraagt tot het slagen van het project.

Indien de student afwezig is op het functioneringsgesprek, of als na het gesprek blijkt dat diens gedrag niet significant is verbeterd, dan krijgt de betrokken

student als quoterings “afwezig” voor de eerste examenkans. Onder een gebrek aan betrokkenheid verstaan we onder andere:

- Niet consequent bijhouden van weekverslagen en tijdregistratie, zelfs na herhaaldelijke feedback hierover door de begeleiders;
- Structureel onvoldoende werkuren registreren;
- Niet kunnen aantonen een significante bijdrage geleverd te hebben aan het project;
- Niet kunnen aantonen dat geregistreerde werkuren ook tot een tastbaar resultaat geleid hebben;
- Meermaals ongewettigd afwezig zijn op de contactmomenten, zonder het team of de begeleiders te verwittigen;
- Niet nakomen van afspraken en beloftes naar de rest van het team of de begeleiders toe.

2.8.2. Tweede examenkans

Wie niet slaagt in de eerste examenkans, krijgt voor de tweede examenkans een persoonlijke opdracht. De opgave wordt via Chamilo gepubliceerd na de bekendmaking van de punten. Wie wil vragen stellen over deze opgave kan dit enkel tijdens de feedback doen. Er geen verdere begeleiding voorzien.

De deadline voor het indienen van deze opdracht is de eerste dag na het zomerreces. Het dossier bestaat uit de GitHub-repository die je toegewezen wordt. Deze bevat naast code en technische documentatie ook een logboek en tijdregistratie. We hebben wat dat betreft dezelfde verwachtingen als in de eerste examenkans. Je houdt dus regelmatig je logboek bij, registreert consequent gepresteerde werkuren en zorgt dat de technische documentatie volledig is. De toestand van je repository op het moment van de deadline wordt beschouwd als je inzending, latere wijzigingen worden niet in rekening gebracht. Je komt een demonstratie geven van het resultaat van deze opdracht op de plaats en tijdstip die in het examenrooster is voorzien. Dat kan op de campus doorgaan (als de corona-maatregelen op dat moment het toelaten) of online via een Teams-gesprek.

Wanneer je echter geen repository aangemaakt hebt, zelf geen code gecommit hebt, of het dossier is onvolledig, dan krijg je als quoterings meteen “afwezig” en volgt er ook geen demonstratie. Je wordt hiervan desgevallend per e-mail op de hoogte gebracht.

Omdat de categorieën “Functioneren binnen het team” en “Retrospectieve” niet kunnen beoordeeld worden in de tweede examenkans, wordt voor deze criteria de beoordeling van de eerste examenkans ongewijzigd overgenomen. Het

gezaamenlijk gewicht van deze criteria is 30% van het totale examencijfer. Een beoordeling “nog niet bekwaam” voor beide betekent in de praktijk dat je nog max. 14/20 kan halen, en enkel op voorwaarde dat je op alle andere criteria het hoogste niveau haalt.

2.8.3. Inhaalkansen

De evaluatiemomenten zijn een groepsgebeuren en de opdracht is geïntegreerd. Het resultaat van je werk individueel tonen is niet mogelijk omdat het afhangt van het werk van het gehele team.

Daarom is het praktisch gezien ook onmogelijk om een inhaalkans te organiseren voor studenten die gewettigd afwezig zijn op een evaluatiemoment.

Een student die tijdens het semester aangetoond heeft de beoogde leerresultaten behaald te hebben, kan in dat geval wel geëvalueerd worden op basis van observatie tijdens de intervisiegesprekken en de schriftelijke neerslag van hun bijdrage. Bij de finale oplevering moeten de componenten waarvoor deze student de verantwoordelijkheid heeft opgenomen kunnen opgezet en gedemonstreerd worden door de andere teamleden.

2.8.4. Dossier

De evaluatie is voor een belangrijk deel gebaseerd op basis van het projectdossier. Dit bestaat voor elke projectteam uit de GitHub-repository die het eindproduct en bijhorende technische documentatie bevat, en ook de weekverslagen en rapporten over tijdregistratie. Ook het Trello-bord van elk team wordt als onderdeel van het dossier beschouwd.

Let op! Aan het einde van het academiejaar worden alle repositories verwijderd. Als je jullie code voor later wil bijhouden, moet je dus zelf tijdig voor een back-up zorgen.

De **team-repository** omvat o.a.:

- Een README met een overzicht van de inhoud van de repository. De begeleiders moeten snel alle nodige info kunnen terugvinden, zoals IP-adrestabel, procedurebeschrijvingen, testplannen, code voor een bepaalde server, enz.
- Algemene info en overzicht: IP-adrestabel, afbeelding van het netwerk, werkverdeling (dit laatste kan via GitHub Issues/Projects of KanBan-bord), enz.;

- De opdrachtomschrijving;
- Het lastenboek voor elke deeltaak:
 - Specificaties en requirements;
 - Verantwoordelijke voor realisatie, verantwoordelijke voor testen;
 - Tijdschatting voor realisatie (in manuur);
- Na realisatie: werkelijk tijdgebruik aanvullen en een verklaring voor het afwijken van de schatting.
- Resultaten van de deeltaken:
 - Broncode (bv. van scripts, geautomatiseerde tests, enz);
 - PacketTracer-bestanden;
 - Enz.
- Technische documentatie, waaronder:
 - Functionele testplannen (test van individuele componenten); m
Draaiboek productie-release;
 - Integratietestplannen (scenario's van gebruikershandelingen waarmee je kan aantonen dat verschillende componenten samenwerken als geheel);
 - Schriftelijke neerslag van opzoekingswerk voor het uitvoeren van een taak;
 - Cheat sheets en procedurebeschrijvingen voor vaak voorkomende taken;
 - Enz.
- Weekrapporten met:
 - Individueel overzicht van gerealiseerde taken met links naar relevante commits/documenten in de team-repo;
 - Individuele timesheets voor de afgelopen periode + overzicht van de hele groep;
 - Screenshot van KanBan-bord op het moment van het intervisiegesprek alsook cummulatief flow schema;
 - Individuele doelstellingen voor de volgende periode;
 - Vragen en/of opmerkingen die het team aan de begeleiders wil stellen.

VM-images en gelijkaardige grote bestanden behoren niet tot het dossier.

Elk teamlid zorgt er voor dat de eigen bijdrage in het weekrapport en tijdregistratie bij aanvang van elk contactmoment volledig up-to-date is. Elke begeleider moet in één oogopslag een beeld kunnen vormen van wat je de afgelopen periode gerealiseerd hebt. Voeg links toe naar je GitHub commits.

Eén van de competenties die we bij het project willen bijbrengen is het bewust worden van je **tijdgebruik**. Jullie hebben al ondervonden dat inschatten hoeveel tijd je aan een taak besteedt niet makkelijk is. Bovendien is het in veel bedrijven gebruikelijk dat je aan je leidinggevende rapporteert over hoe je je tijd gebruikt. Dit kan al eenvoudig door elke dag consequent te noteren wat je doet voor het project en voor hoe lang. Er bestaan ook specifieke tools voor het bijhouden van timesheets en het maken van overzichten. Spreek onderling en met de begeleider voor Analyse af op welke manier jullie dit doen en welke tool(s) jullie zullen gebruiken.

Breng elke week een overzicht mee naar het opvolgingsgesprek waarin duidelijk is hoeveel tijd je gespendeerd hebt aan het project. Tel daarbij ook zaken mee die buiten de opgegeven taken liggen. Dit zet je onder de noemer “Overhead.” Overhead omvat o.a. projectadministratie, opzetten werkomgeving, volgen online cursus, ...

Aan het einde van het project moet je in **totaal 200 werkuren** kunnen inbrengen, wat de totale begrote studietijd is volgens de studiefiche. Verdeeld over 13 weken komt dat op ongeveer 15uur per week. Dat omvat zowel de tijd van alle contactmomenten opgeteld (incl. de kick-off!), als de tijd die je daarbuiten hebt gespendeerd aan het project (info opzoeken, individueel werken aan de opdracht, teamoverleg buiten de contactmomenten, enz.).

Omdat dit een verderzetting is van de werkwijze van vorig jaar, gaan we er van uit dat iedereen zich hier meteen aan houdt, zodat de gesprekken over inhoudelijke zaken kunnen gaan en dat we niet opnieuw moeten uitleggen wat we van jullie verwachten qua rapportering.

3

Opgave

Met dit project probeer je aan te tonen dat je in staat bent om met je team complexe ICT-infrastructuur geautomatiseerd kan opzetten en kan laten functioneren. De bedoeling van het project is om een volledig functioneel netwerk op te zetten met alle typische services: DNS, centraal gebruikersbeheer, DHCP, enz. Initieel in een “on-premise” opstelling, daarna in een “hybrid cloud” omgeving.

Het is aan het team om te beslissen wie welke taken op zich zal nemen om die uit te werken, te testen en op te leveren. Elk teamlid draagt de eindverantwoordelijkheid voor minstens één deeltaak/component van het netwerk en beschrijft dit in haar/zijn logboek.

Lees de opgave grondig en doe zo goed mogelijk wat gevraagd wordt. Let er op dat bij verschillende taken addertjes onder het gras zitten, of dat ze bewust vaag geformuleerd zijn. Waar er geen expliciete keuze is opgelegd, kan het team zelf beslissen, in samenspraak met de begeleiders. De opgave kan in de loop van het semester, naargelang de omstandigheden, nog bijgestuurd worden. De begeleiders kunnen bijkomende requirements opleggen of desgevallend de scope beperken. Het team kan zelf ook initiatief nemen om - telkens in samenspraak met de begeleiders- extra's te implementeren.

Jullie zullen merken dat jullie bij de meeste opdrachten van elkaar afhangen. Maak dus duidelijke afspraken die voor iedereen toegankelijk zijn via de technische documentatie van het project. Streef er naar om diegenen die van jou afhangen zo goed mogelijk te helpen en hun werk zo vlot mogelijk te maken. Dat kan bestaan uit het ter beschikking stellen van een testomgeving voor de componenten onder jouw verantwoordelijkheid, hulp bij het gebruik ervan, of het vereenvoudigen van het gebruik door automatisering.

3.1. Algemeen

- Het domein zal de volgende naam hebben: **"thematrix.local"**.
- Leg eerst en vooral een IP-adrestabel vast voor alle componenten in het netwerk die dit nodig hebben. Plan vooraf de nodige subnetten en verspil geen IP-adressen.
- De serverinfrastructuur bestaat uit virtuele machines gebaseerd op een evenredige mengeling van Windows Server 2019 en de laatste versie van AlmaLinux (versie 9.1 op 8 februari 2023).
- Installatie en configuratie van de Windows systemen gebeurt deels in de GUI en deels aan de hand van PowerShell scripts. Op de Linux-systemen gebeurt dit volledig met Vagrant en bash-scripting.
- Containertechnieken zoals docker, docker-compose, podman, LXC, ... zijn niet toegelaten.
- Zorg ervoor dat alle scripts herbruikbaar zijn. Hiermee bedoelen we dat je "hard-coded" waarden vermijdt, maar in plaats daarvan overal variabelen gebruikt. De systeembeheerder kan de gewenste waarden invullen in een configuratiebestand dat ingelezen wordt door je scripts, of (in het geval van een PowerShell-script) via een dialoogvenster bij de uitvoering van het script.
- Maak voor de Windows-installaties *geen* gebruik van Vagrant. Volgens de documentatie is dit wel degelijk mogelijk, maar in de praktijk is dit erg omslachtig en zullen jullie hier veel tijd mee verliezen.
- Schrijf testplannen met exacte procedures die toelaten te valideren of een deeltaak is uitgevoerd volgens de specificaties.
- Een ander teamlid volgt de instructies van de testplannen en schrijft een testrapport over het resultaat. Indien er tests falen, wordt een ticket aangemaakt in Trello (met een link naar het rapport) zodat de verantwoordelijke de fout kan oplossen.

3.2. Netwerk

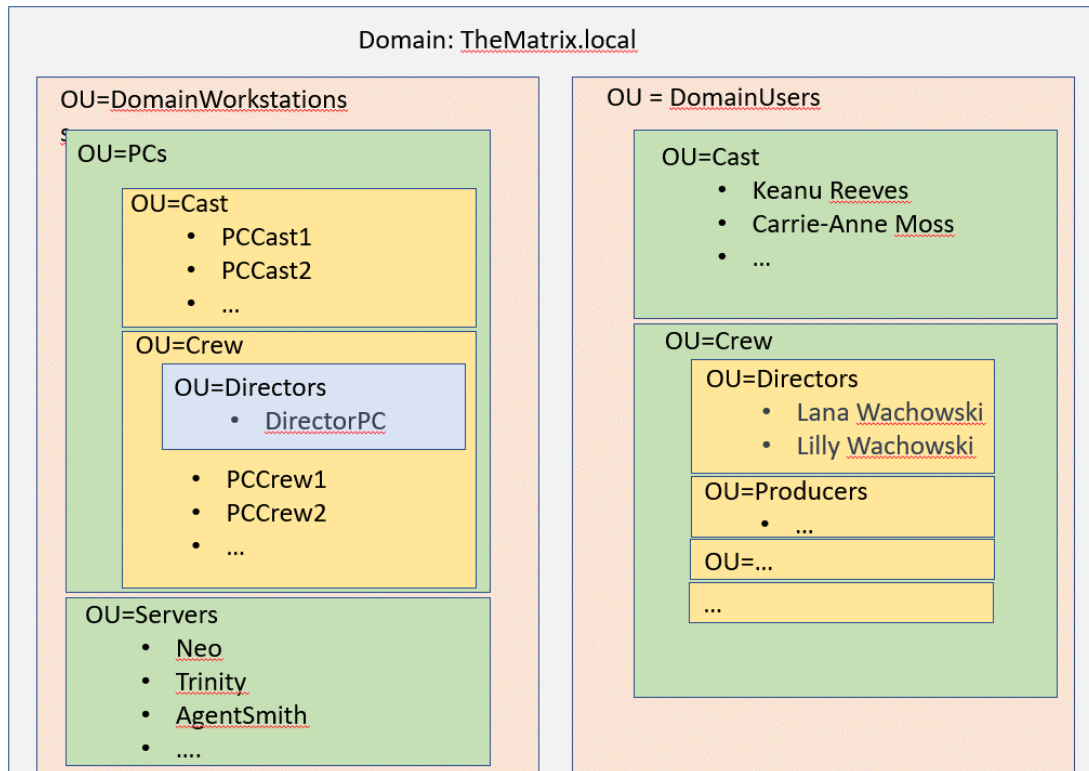
- Het netwerk en alle servers worden uitgewerkt met IPv4. IPv6 wordt nog niet uitgerold op de apparatuur maar wordt wel al voorzien op de DNS-server (zie verder).
- De gehele opstelling wordt lokaal uitgevoerd met virtuele machines en de aanwezige apparatuur in het netwerkklokaal.
- Simuleer de netwerkinfrastructuur met Packet Tracer¹.
 - Voorzie VLANs voor de cast en de crew:

¹ <https://www.netacad.com/portal/resources/packet-tracer>

- VLAN 20 Interne servers
 - Vaste, private IP-adressen
 - De IP-adressen corresponderen met de adressering van de servers.
- VLAN 30 Werkstations crew
 - Dynamische, private IP-adressen (via DHCP)
 - Kunnen interne servers en Internet bereiken
- VLAN 40 Werkstations cast
 - Dynamische, private IP-adressen (via DHCP)
 - Kunnen enkel op Internet, interne servers zijn niet bereikbaar
- Inter-VLAN routing wordt uitgevoerd met een router-on-a-stick configuration
- Toegang tot internet gebeurt via configuratie van NAT.
- Na succesvolle simulatie worden dezelfde configuraties geïmporteerd op de apparatuur in het netwerklokaal, en worden de (virtuele) servers aangesloten tot één werkend geheel. De virtuele machines draaien op de laptops van de studenten. Er zullen soms meerdere VM's op een enkele laptop moeten draaien, denk goed op voorhand na over deze verdeling voor om performantiebottlenecks en andere problemen te vermijden.

3.3. Domain Controller

- Hostnaam: “**agentsmith**”
- OS: Windows Server 2019 (Geen GUI, enkel CLI)
- Implementeer volgende domainstructuur automatisch via een powershellscript:

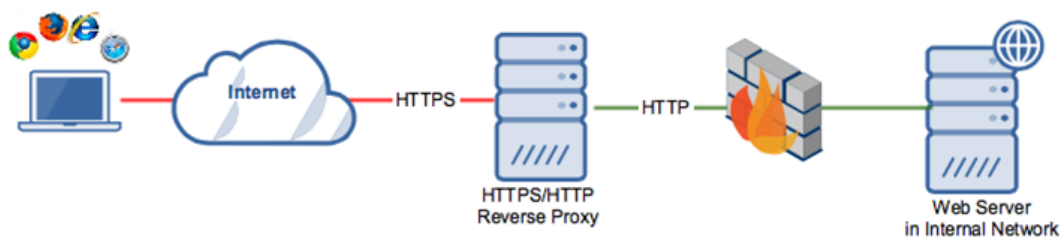


- Importeer met een powershellscript ook enkele acteurs, crewleden en minstens 2 PC's in elke OU vanuit een CSV-file.
- Zorg ervoor dat je de servers via RSAT kan configureren vanop de DirectorPC (Windows 10).
- Pc's en servers hebben geen eigen gebruikers, authenticatie gebeurt telkens via de Domain Controller.
- Zorg ervoor dat de cast enkel kan inloggen in de cast-Pc's, de crew enkel op de crew-Pc's. Directors kunnen overal inloggen.
- Elke user krijgt automatisch zijn eigen persoonlijke shared folder.
- Voorzie ook een shared folder voor zowel de cast als de crew en creëer groepen om de toegang tot die shared folders te regelen.
- Werk volgende beleidsregels uit op gebruikersniveau:
 - verbied iedereen behalve de directors de toegang tot het control panel
 - verwijder het games link menu uit het start menu voor iedereen
 - verbied iedereen van de cast de toegang tot de eigenschappen van de netwerkadapters.

- Deze server zal ook de rol van DNS-server vervullen:
 - Hij beantwoordt alle queries binnen het domein “thematrix.local”.
 - Queries voor andere domeinen worden geforward naar een geschikte DNS-server.
 - Voor elke host binnen het domein zijn er A (IPv4), AAAA (IPv6) en PTR (IPv4 en (!) IPv6) records voorzien, in de gepaste zonebestanden.
 - Voorzie voor elke host geschikte CNAME-records om de functie van een server aan te duiden (bv. www, imap, smtp, ...).
 - Voorzie waar nuttig/nodig ook andere records (bv. NS, MX, SRV, ...).

3.4. Webserver

- Hostnaam: “**trinity**”.
- OS: AlmaLinux.
- Benader de server via SSH: zorg ervoor dat je nooit met het root account kan inloggen en dat je enkel (!) door middel van ssh-keys kan inloggen (dus niet door een gebruikersnaam en wachtwoord in te geven).
- Installeer een webserver naar keuze (bv. Nginx, Apache, ...) en configureer ondersteuning voor HTTPS en HTTP/2. Zorg ervoor dat als de webserver gescand wordt met nmap, deze geen informatie geeft over of de versie ervan. Wat moeilijker is het verbergen van het type webserver, zoek dit uit zodat nmap geen of het verkeerde type webserver weergeeft.
- Installeert daarnaast ook een databank naar keuze (bv. MariaDB, PostgreSQL, ...).
- Installeer tenslotte nog een CMS (content management system) naar keuze (bv. Drupal, Wordpress, ...).
- Je kan vanop elk werkstation naar deze webserver surfen met of zonder het “www” voorvoegsel (<https://www.thematrix.local/> en <https://thematrix.local/>).
- HTTPS gaan we hierbij niet instellen op de webserver. Installeer een reverse proxy naar keuze op de webserver die het https-verkeer regelt (normaal heeft deze zijn eigen server, maar omdat we enkel reverse proxy hanteren is dat in deze situatie overkill). De webserver is dus niet rechtstreeks van buitenaf bereikbaar. Maak gebruik van self-signed certificates voor HTTPS



- Installeer Rallly (<https://github.com/lukevella/rallly>) op de server. Zorg ervoor dat er naar Rallly gesurfd kan worden via <https://www.rallly.thematrix.local/> en <https://rallly.thematrix.local/> . Zorg ervoor dat beide websites tegelijk bereikbaar zijn. De reverse proxy is hiervoor zeer handig, dus maak er gebruik van. Je hoeft geen e-mailondersteuning voor Rallly in te stellen.

3.5. Email

- hostnaam: **"neo"**
- OS: Windows Server 2019 (zonder GUI, enkel CLI)
- Installeer Exchange 2016 / 2019.
- Accounts worden gesynchroniseerd met Active Directory.
- Zorg voor voldoende beveiliging aan de hand van een spam filter en virusscanner.

3.6. Matrix.org server

- Hostnaam: **"theoracle"**
- OS: AlmaLinux
- Draai op deze server Synapse (<https://matrix.org/docs/projects/server/synapse>). Dit is een server voor het matrix.org protocol dat in de open source community steeds meer IRC en andere chatsystemen vervangt. Je kan er makkelijk bots op programmeren en bridges installeren naar andere chat (Discord, Messenger, ...). Daarnaast zijn chats default geëncrypteerd en onleesbaar voor admins.
- Matrix ondersteunt federatie (wat is dit?). Meestal is dit gewenst, maar hier hoef je dit niet op te zetten.
- Maak minstens 2 accounts en voer een geëncrypteerd gesprek.
- Maak een bashscript op de webserver dat een bericht stuurt naar een room als de webserver afsluit. Installeer dit script met een systemd unit en systemd timer.

3.7. Eigen Keuze

- Hostnaam: **"dozer"**
- OS: AlmaLinux

- Draai op deze server een service naar keuze. Je kan inspiratie opdoen uit de meest bekende lijst uit de self-hosted community: <https://github.com/awesome-selfhosted/awesome-selfhosted>. Kies een service die nog niet is opgezet in andere vakken. Hoe meer werk er wordt voor verricht, hoe meer dit wordt beloond. Maak je er hier niet te snel van af, want dat kan ook een negatieve impact hebben.

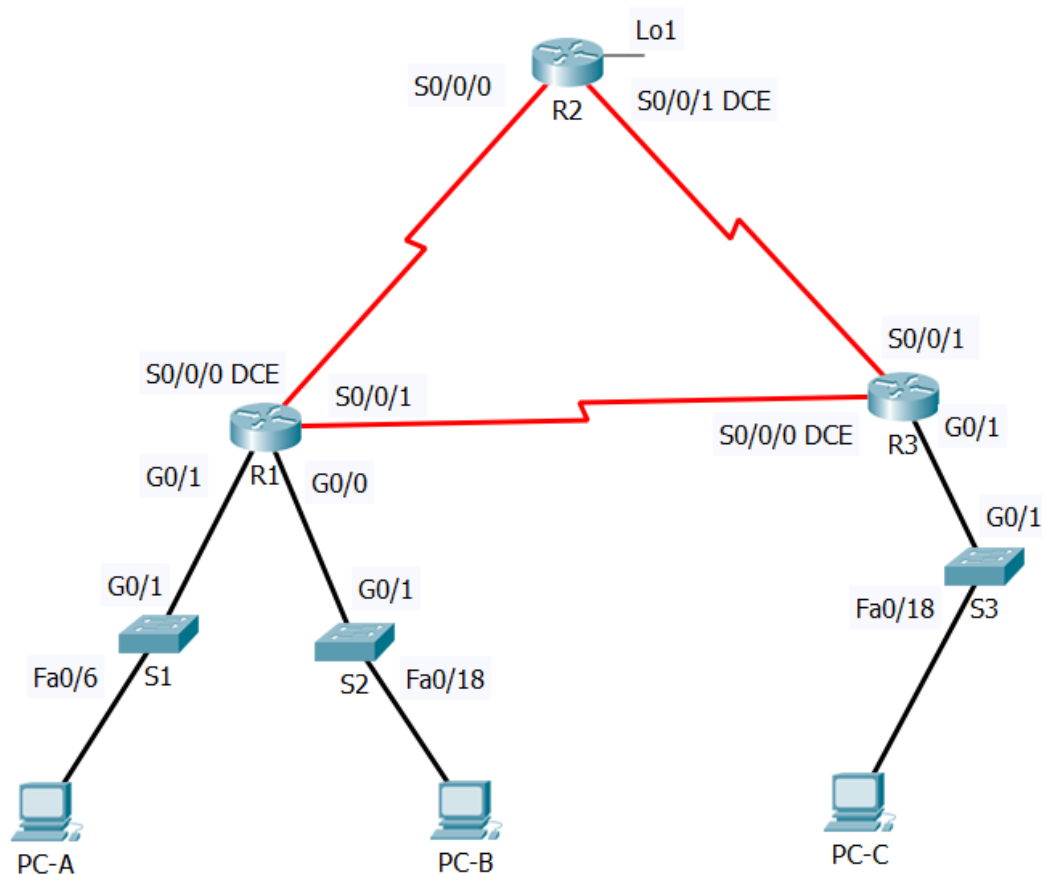
3.8. Cisco labo's

Hierna zijn er 2 Cisco labo's beschreven die jullie in Packet Tracer voorbereiden. Deze worden nadien uitgevoerd en gedemonstreerd op de toestellen in het netwerklokaal. De roostering voor elke groep vind je op Chamilo terug.

3.8.1 Labo 1

Gegeven:

Volgende opstelling



Bijbehorende adressentabel:

Device	Interface	IP Address	Default Gateway
R1 (FE80::1)	S0/0/0(DCE)	2001:DB8:AAAA:1::1/64	N/A
	S0/0/1	2001:DB8:AAAA:3::1/64	N/A
	G0/0	2001:DB8:ACAD:B::1/64	N/A
	G0/1	2001:DB8:ACAD:A::1/64	N/A
R2 (FE80::2)	S0/0/0	2001:DB8:AAAA:1::2/64	N/A
	S0/0/1(DCE)	2001:DB8:AAAA:2::2/64	N/A
	Lo1	2001:DB8:AAAA:4::1/64	N/A
R3 (FE80::3)	S0/0/0(DCE)	2001:DB8:AAAA:3::2/64	N/A
	S0/0/1	2001:DB8:AAAA:2::1/64	N/A
	G0/1	2001:DB8:ACAD:C::1/64	N/A
S1	VLAN1	2001:DB8:ACAD:A::A/64	N/A
S2	VLAN1	2001:DB8:ACAD:B::A/64	N/A
S3	VLAN1	2001:DB8:ACAD:C::A/64	N/A
PC-A	NIC	2001:DB8:ACAD:A::3/64	FE80::1
PC-B	NIC	2001:DB8:ACAD:B::3/64	FE80::1
PC-C	NIC	2001:DB8:ACAD:C::3/64	FE80::3

DEEL 1: maak de netwerkopstelling en initialiseer de toestellen

Stap 1: bouw de netwerkopstelling uit overeenkomstig de gegeven topologie

Stap 2: initialiseer of reload eventueel de routers en switchen: zorg er dus voor dat er geen oude configuratie meer op de toestellen staat

DEEL 2: configureer alle toestellen en controleer de verbindingen

Stap 1: configureer de IPv6 adressen op alle PC's

Configureer de IPv6 global unicast adressen overeenkomstig de gegeven adressentabel. Gebruik het link-local adres als default-gateway op alle Pc's.

Stap 2: configureer de switchen

- a. Maak DNS lookup ongedaan.
- b. Configureer een hostname.
- c. Wijs volgende domeinnaam toe: **ccna-lab.com**.
- d. Encrypteer de plain-text paswoorden.
- e. Maak een MOTD-banner die de gebruikers waarschuwt: "Toegang voor onbevoegden is verboden".
- f. Maak een lokale user database met een gebruikersnaam **admin** en paswoord **classadm**.
- g. Configureer **class** als het privileged EXEC geëncrypteerd paswoord.
- h. Configureer **cisco** als het console paswoord en maak login mogelijk.
- i. Maak login op de VTY-lijnen mogelijk door gebruik te maken van de lokale database.
- j. Genereer een crypto rsa key voor ssh, gebruik makend van een modulus grootte van 1024 bits.
- k. Verander de transport input op alle VTY-lijnen naar alleen SSH en Telnet.
- l. Wijs een IPv6 adres toe aan VLAN 1 overeenkomstig de adrestabel.

Voeg hier tussen de runningconfiguration file van S1.

Stap 3: configureer de basisinstellingen op alle routers

- a. Maak DNS lookup ongedaan.
- b. Configureer een hostname.
- c. Wijs volgende domeinnaam toe: **ccna-lab.com**.
- d. Encrypteer de plain-text paswoorden.
- e. Maak een MOTD-banner die de gebruikers waarschuwt: "Toegang voor onbevoegden is verboden".
- f. Maak een lokale user database met een gebruikersnaam **admin** en paswoord **classadm**.
- g. Configureer **class** als het privileged EXEC geëncrypteerd paswoord.
- h. Configureer **cisco** als het console paswoord en maak login mogelijk.

- i. Maak login op de VTY-lijnen mogelijk door gebruik te maken van de lokale database.
- j. Genereer een crypto rsa key voor ssh, gebruik makend van een modulus grootte van 1024 bits.
- k. Verander de transport input op alle VTY-lijnen naar alleen SSH en Telnet.

Stap 4: configureer IPv6 instellingen op R1

- a. Configureer de IPv6 unicast adressen op de volgende interfaces: G0/0, G0/1, S0/0/0 en S0/0/1.
- b. Configureer de IPv6 link-local adressen op de volgende interfaces: G0/0, G0/1, S0/0/0 en S0/0/1. Gebruik **FE80::1** voor de link-local adressen op alle vier interfaces.
- c. Zet de clock rate op S0/0/0 op 128000.
- d. Zorg ervoor dat de interfaces IPv6-pakketten kunnen versturen.
- e. Maak IPv6 unicast routing mogelijk:

```
R(config)# ipv6 unicast-router
```

- f. Configureer OSPFv3 op R1 en zorg dat de LAN-interfaces passieve interfaces zijn.

- Configuratie OSPFv3:

```
R(config)# ipv6 router ospf 10
R(config-rtr)# passive interface G0/0/0 (indien G0/0/0 de
passieve interface is)
```
- Dan op elke actieve interface:

```
R(config-if)#ipv6 ospf 10 area 0
```

Voeg hier tussen de runningconfiguration file van R1.

Stap 5: configureer IPv6 instellingen op R2

- a. Configureer de IPv6 unicast adressen op de volgende interfaces: Lo1, S0/0/0 en S0/0/1.
- b. Configureer de IPv6 link-local adressen op de volgende interfaces: S0/0/0 en S0/0/1. Gebruik **FE80::2** voor de link-local adressen op alle twee interfaces.
- c. Zet de clock rate op S0/0/1 op 128000.
- d. Zorg ervoor dat de interfaces IPv6-pakketten kunnen versturen.
- e. Maak IPv6 unicast routing mogelijk.

f. Maak een default route die gebruik maakt van de loopback interface Lo1 (deze dient ter simulatie van een internetconnectie).

```
R(config)# ipv6 route ::/0 Lo1
```

g. Configureer OSPFv3 op R2 en zorg dat de default route doorgegeven wordt op de andere routers van het domein.

- Configuratie zie R1 en voeg een lijn toe, onder:

```
R(config)# ipv6 router ospf 10
R(config-rtr)# passive interface G0/0/0 (indien G0/0/0 de passieve interface is)
R(config-rtr)# default-information originate
```

Voeg hier tussen de runningconfiguration file van R2.

Stap 6: configureer IPv6 instellingen op R3

- Configureer de IPv6 unicast adressen op de volgende interfaces: G0/1, S0/0/0 en S0/0/1.
- Configureer de IPv6 link-local adressen op de volgende interfaces: G0/1, S0/0/0 en S0/0/1. Gebruik **FE80::3** voor de link-local adressen op alle drie interfaces.
- Zet de clock rate op S0/0/0 op 128000.
- Zorg ervoor dat de interfaces IPv6-pakketten kunnen versturen.
- Maak IPv6 unicast routing mogelijk.
- Configureer OSPFv3 op R3 en maak van de LAN-interface een passieve interface.

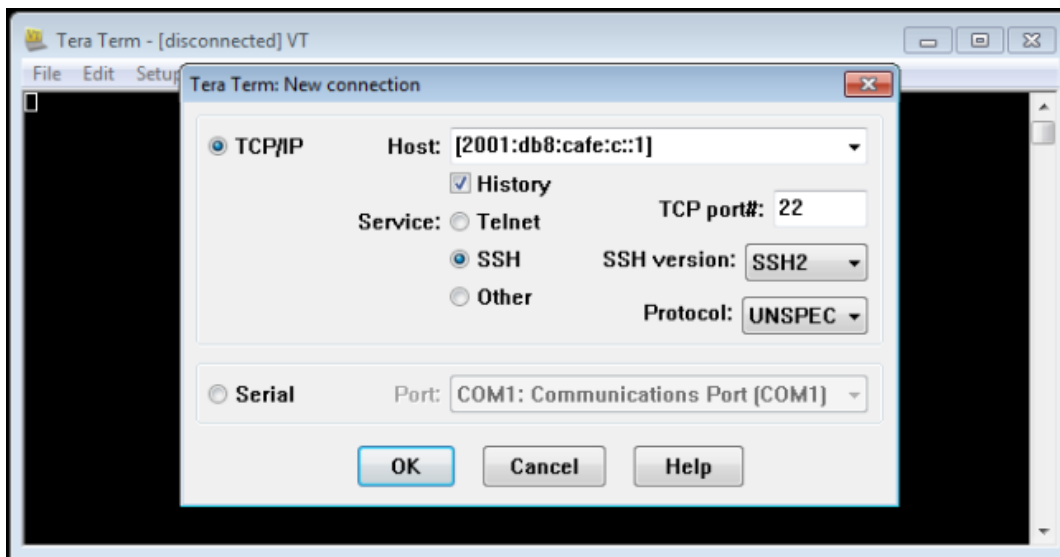
- Configuratie zie R1

Voeg hier tussen de runningconfiguration file van R3.

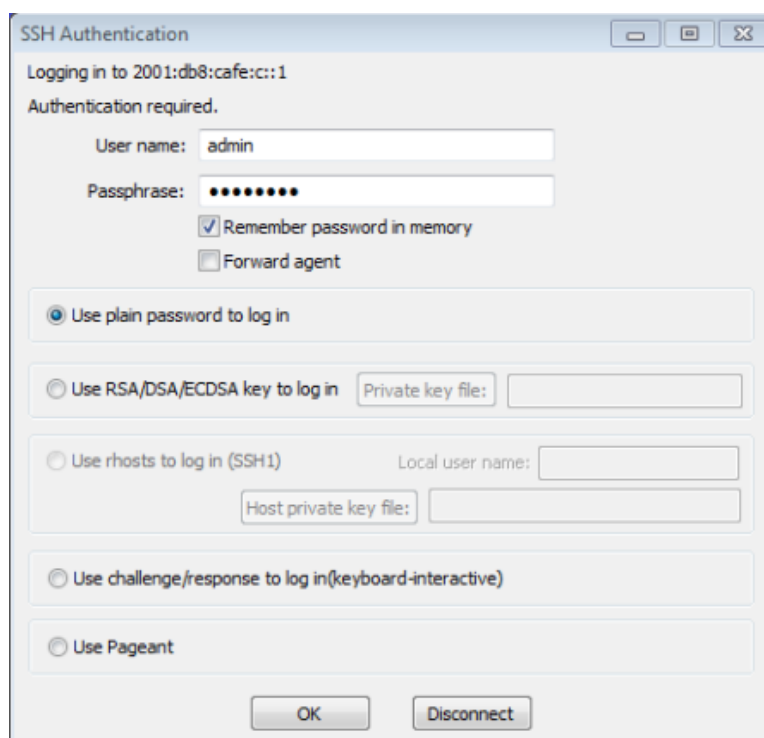
Stap 7: Controleer de connectiviteit

- Elke PC zou in staat moeten zijn om te pingen naar de andere PC's in de topology.
- Telnet naar R1 vanuit alle PC's in de topology.
- SSH naar R1 vanuit alle PC's in de topology.
- Telnet naar S1 vanuit alle PC's in de topology.
- SSH naar S1 vanuit alle PC's in de topology.
- Zorg ervoor dat nu alles werkt: troubleshoot indien nodig want de ACLs die je zal maken in DEEL3 gaan de toegang beperken tot bepaalde zones van het netwerk

Opmerking: Tera Term vereist dat het bestemmings IPv6 adres tussen vierkante haakjes staat. Voer het IPv6 adres in zoals in de figuur, klik op **OK** en vervolgens op **Continue** om de beveiligingswaarschuwing te accepteren en te connecteren met de router.



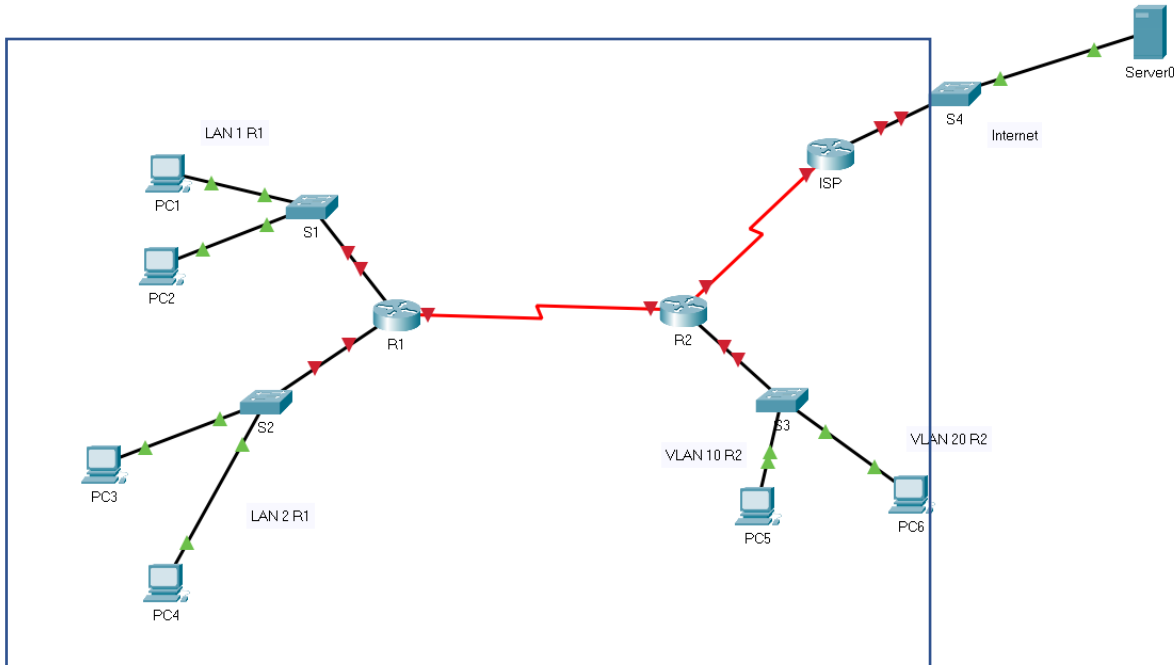
Voer de geconfigureerde gebruikerscredentials in (gebruikersnaam **admin** en paswoord **classadm**) en selecteer de **Use plain password to log in** in de SSH Authentication dialogue box. Klik op **OK** om verder te gaan.



3.8.2 Labo 2

Gegeven:

Grondplan



Ter voorbereiding:

Gebruik de bijbehorende Packet Tracer file: **Opgave DHCPv6 met statische routes klaar voor configuratie.pkt** (Zie Chamilo)

Labo-opdracht :

Bouw het netwerk binnen de kader uit. Vervang hierbij de interface naar het internet of dus Server0 door een **loopback interface** op de ISP-router.

Opdracht:

1. Te gebruiken netwerkadressen zijn terug te vinden in de tabel hieronder.
2. R1:
 - a. Link local adres op alle interfaces = FE80::1
 - b. Configureer alle interfaces eerst. Neem telkens als interface ID = 1, behalve S0/1/0
 - c. G0/0/0 wordt geactiveerd voor SLAAC => PC1 en PC2 krijgen hun IPv6-adres via SLAAC
 - d. Via G0/0/1 wordt R1 gebruikt als stateless DHCPv6-server:
 - i. Domainname : SystemEngineeringProject
 - ii. DNS-server: 2001:db8:1000::10

- iii. => PC3 en PC4 krijgen hun IPv6-adres via de stateless DHCPv6-server
 - e. S0/1/0 is client van de stateful DHCPv6-server op R2.
- 3. R2:
 - a. Link local adres op alle interfaces = FE80::2
 - b. Configureer alle interfaces eerst. Neem telkens als interface ID = 1.
 - i. Let op G0/0/0 bedient 2 VLAN's.
 - 1. Voorzie de nodige subinterfaces
 - ii. Voorzie intervlan-routing via router-on-a-stick
 - c. Via G0/0/0 en S0/1/0 wordt R2 geactiveerd als stateful DHCPv6 voor
 - i. VLAN 10 waarop PC5 zit (VLAN 10 zit op interfaces fa0/1-fa0/10)
 - ii. VLAN 20 waarop PC6 zit (VLAN 20 zit op interfaces fa0/11-fa0/20)
 - iii. Voor de seriële verbinding met R1, want R1 is client van deze stateful DHCPv6
 - iv. Domainname : SystemEngineeringProject
 - v. DNS-server: 2001:db8:1000::10
- ➔ Tip voorzie voor elk netwerk een aparte DHCP-pool!
- 4. Routing:
 - a. Voorzie op R1 een fully specified statische default route door gebruik te maken van het link-local adres van R2
 - b. Voorzie op R2:
 - i. 2 directly connected standaard statische routes
 - 1. 1 naar netwerk van LAN 1 R1
 - 2. En 1 naar netwerk van LAN 2 R1
 - ii. Een next hop statische default route naar de ISP.
 - c. Voorzie op de ISP-router 4 standaard statische routes (naar keuze) naar:
 - i. 1 naar netwerk van LAN 1 R1
 - ii. 1 naar netwerk van LAN 2 R1
 - iii. 1 naar VLAN 10 R2
 - iv. 1 naar VLAN 20 R2
- 5. Controleer of je op alle PC's de webpagina: systemengineeringproject.org kan oproepen!

Gebruik alleen de IP netwerken die achter jullie groep staan.

Groep	LAN 1 R1/64	LAN 2 R1/64	VLAN 10 R2/64	VLAN 20 R2/64	Seriële link R1-R2/64
Groep 1 Aalst	2001:db8:A:1::	2001:db8:B:1::	2001:db8:C:1::	2001:db8:D:1::	2001:db8:ACDC:1::
Groep 1 TIAO	2001:db8:A:2::	2001:db8:B:2::	2001:db8:C:2::	2001:db8:D:2::	2001:db8:ACDC:2::
Groep 2 TIAO	2001:db8:A:3::	2001:db8:B:3::	2001:db8:C:3::	2001:db8:D:3::	2001:db8:ACDC:3::
Groep 1 Gent	2001:db8:A:4::	2001:db8:B:4::	2001:db8:C:4::	2001:db8:D:4::	2001:db8:ACDC:4::
Groep 2 Gent	2001:db8:A:5::	2001:db8:B:5::	2001:db8:C:5::	2001:db8:D:5::	2001:db8:ACDC:5::
Groep 3 Gent	2001:db8:A:6::	2001:db8:B:6::	2001:db8:C:6::	2001:db8:D:6::	2001:db8:ACDC:6::
Groep 4 Gent	2001:db8:A:7::	2001:db8:B:7::	2001:db8:C:7::	2001:db8:D:7::	2001:db8:ACDC:7::
Groep 5 Gent	2001:db8:A:8::	2001:db8:B:8::	2001:db8:C:8::	2001:db8:D:8::	2001:db8:ACDC:8::
Groep 6 Gent	2001:db8:A:9::	2001:db8:B:9::	2001:db8:C:9::	2001:db8:D:9::	2001:db8:ACDC:9::

Groep 7 Gent	2001:db8:A:10::	2001:db8:B:10::	2001:db8:C:10::	2001:db8:D:10::	2001:db8:ACDC:10::
Groep 8 Gent	2001:db8:A:11::	2001:db8:B:11::	2001:db8:C:11::	2001:db8:D:11::	2001:db8:ACDC:11::
Groep 9 Gent	2001:db8:A:12::	2001:db8:B:12::	2001:db8:C:12::	2001:db8:D:12::	2001:db8:ACDC:12::
Groep 10 Gent	2001:db8:A:13::	2001:db8:B:13::	2001:db8:C:13::	2001:db8:D:13::	2001:db8:ACDC:13::
Groep 11 Gent	2001:db8:A:14::	2001:db8:B:14::	2001:db8:C:14::	2001:db8:D:14::	2001:db8:ACDC:14::
Groep 12 Gent	2001:db8:A:15::	2001:db8:B:15::	2001:db8:C:15::	2001:db8:D:15::	2001:db8:ACDC:15::
Groep 13 Gent	2001:db8:A:16::	2001:db8:B:16::	2001:db8:C:16::	2001:db8:D:16::	2001:db8:ACDC:16::
Groep 14 Gent	2001:db8:A:17::	2001:db8:B:17::	2001:db8:C:17::	2001:db8:D:17::	2001:db8:ACDC:17::
Groep 15 Gent	2001:db8:A:18::	2001:db8:B:18::	2001:db8:C:18::	2001:db8:D:18::	2001:db8:ACDC:18::

4

Tools, teamwork

In dit project zullen jullie samenwerken in een groter team, dan tot nu toe het geval geweest is. Dat maakt dat jullie je zullen moeten aanpassen en heel goede afspraken maken om conflicten (hetzij in de code op het versiebeheersysteem, hetzij tussen teamleden) te vermijden.

Hierna volgen enkele aanbevelingen.

Spreek goed af binnen je team op welke manier jullie te werk zullen gaan, zorg er voor dat dit duidelijk gedocumenteerd is en dat deze documentatie voor alle teamleden en begeleiders vlot toegankelijk is.

4.1. Professioneel gebruik van Git

Wanneer je met meerdere personen samenwerkt aan een gemeenschappelijke codebase, vergroot de kans op merge-conflicten. Er zijn verschillende strategieën om dit te vermijden. Je kan als team zelf beslissen hoe je dit gaat aanpakken.

4.1.1. Voorbereiding

- Gebruik een geschikte **GitHub-gebruikersnaam** die je ook na je afstuderen nog kan gebruiken. Je HOGENT-login (van de vorm 123456ab) is niet geschikt. Ook puberale of aanstootgevende nicknames vermijd je best.

Je kan indien nodig een nieuw account aanmaken of in je bestaande account je gebruikersnaam wijzigen (klik op je avatar rechtsboven > Settings > Account > Change username).

- Registreer je HOGENT **studenten-emailadres** in je GitHub account (en voor na je afstuderen liefst ook een privé-adres). Via je studenten-emailadres kan je aanspraak maken op het GitHub Student Developer Pack¹.
- Genereer een **SSH sleutelpaar** als je dat nog niet gedaan hebt, en registreer de publieke sleutel op GitHub zodat je op een eenvoudigere manier kan synchroniseren met GitHub.
- Zorg dat je naam en HOGENT-emailadres zeker **geconfigureerd** zijn in de **Git client**. Als je dit niet doet, dan wordt het ook moeilijk je commits terug te vinden in de historiek en kan je je bijdrage aan het project niet aantonen. Het is dus erg belangrijk om al van in het begin correct te doen!

```
$ git config --global user.name "An Pieters"
$ git config --global user.email
"an.pieters@student.hogent.be"
```

4.1.2. Algemene richtlijnen

Eerst en vooral is een goede, **overzichtelijke directorystructuur** belangrijk. Mergeconflicten komen vooral voor wanneer verschillende personen tegelijk hetzelfde bestand bewerken. Als je goede afspraken maakt over wie welke bestanden bewerkt, vermijd je al veel problemen.

Goede commit-boodschappen zijn ook des te belangrijker om aan je teamleden te communiceren wat je precies gedaan hebt². Aan boodschappen als "wijzigingen", "fix", "herwerken", "brol", "plz work", enz. heeft niemand iets: noch de begeleiders, noch je teamleden, noch je toekomstige zelf. Je kan afspraken maken over prefixen die aangeven aan welke taak je gewerkt hebt, bv. de hostnaam van de server waaraan je gewerkt hebt: "[bravo]", of

¹ <https://education.github.com/pack>

² <https://chris.beams.io/posts/git-commit/>

algemene aanduidingen als “[doc]”, “[fix]”, enz. Het nummer van de Git issue kan ook informatief zijn, bv. “[bravo #13]”, “[doc #36]”, “[fix #42]”, enz. Het laatste voorbeeld zal ook Issue #42 sluiten (zorg dus dat je zeker bent dat de issue wel degelijk opgelost is!).

Hou commits zo klein mogelijk. Maak zeker niet de fout om slechts één of enkele keren per week een grote commit uit te voeren. Wacht ook niet tot je een component volledig afgewerkt hebt, maar registreer ook deelresultaten. Hoe groter een commit en hoe langer je code niet gesynchroniseerd is met de centrale Git repository, des te meer kansen op merge-conflicten. Commit dus meerdere keren per werksessie, en telkens je één concrete stap vooruit raakt.

Synchroniseer ook heel regelmatig met de centrale Git repository. Minstens aan het einde van elke sessie dat je aan het project werkt, maar vaker mag zeker.

Overschrijf nooit publieke historiek, en gebruik dus nooit de volgende commando's:

```
# Doe het volgende NOOIT!:  
$ git reset --hard  
$ git push --force
```

Dit zal voor alle teamleden leiden tot conflicten en mogelijks tot verlies van geleverd werk. Als je in je eigen werkkopie van de repository terug wil naar de toestand in een vorige commit, gebruik je “git revert”.

4.1.3. Branching & merging, Pull Requests

In vele teams wordt gebruik gemaakt van *branches* om individuele bijdragen van teamleden gecontroleerd toe te voegen aan de gemeenschappelijke code base. Teamleden werken dan typisch *nooit* op de master branch. De werkwijze is dan zo:

```
$ git pull  
$ git checkout --branch feature/newstuff  
# ... wijzigingen ...  
$ git add .
```

```
$ git commit --message "Beschrijvende boodschap"
$ git push origin feature/newstuff
```

Op de tweede lijn wordt een nieuwe zgn. *topic branch* gemaakt met de naam "feature/newstuff". Over branch-namen worden typisch ook goede afspraken gemaakt binnen een team. Bijvoorbeeld dat de branch-naam begint met een prefix dat aangeeft wat voor soort wijziging het betreft, bv. "feature/", "fix/" (bug oplossen), "doc/", enz. Vaak wordt er ook verplicht om het nummer van de issue/ticket toe te voegen aan de branch-naam, bv. "feature/22", "feature/22-dns-server", of "fix/56-dhcp-wont-start".

Na het pushen naar de centrale Git repository, kan dan verder gewerkt worden aan een ander ticket. De volgende stap is er dan voor zorgen dat alle wijzigingen in de centrale repository samengevoegd worden (*merge*) met de master branch. In de praktijk worden enkele teamleden verantwoordelijk gesteld voor het integreren van de code. GitHub kan daarbij helpen en zal typisch aangeven of een branch kan samengevoegd worden met master. Het mergen is dan kwestie van op een knop te klikken, een beschrijving toe te voegen en een commitboodschap toe te voegen.

De voordelen van deze werkwijze zijn o.a.:

- Het is makkelijker te garanderen dat de master-branch steeds een werkende versie van het project bevat.
- Aan het mergen kan er een Q/A-proces gekoppeld worden dat introductie van fouten vermijdt. Een tester kan het testplan uitvoeren en pas als dit slaagt de branch mergen (met toevoeging van het testrapport, uiteraard). Je kan ook "commit hooks" toevoegen die bij elke commit geautomatiseerde testen uitvoeren, een linter om codestijl te controleren, of een andere Q/A-tool.
- Wanneer verschillende teamleden tegelijk wijzigingen willen pushen (bv. op het einde van een van de contactmomenten), is het makkelijker deze gecontroleerd en één voor één te mergen.

Ook hier zijn echter gevaren aan verbonden:

- Deze werkwijze is een stuk complexer, en elk teamlid moet zich strikt aan de afspraken houden.
- De verantwoordelijke(n) voor de integratie van code is/zijn een bottleneck op de vooruitgang van het project als geheel.

- Er kunnen zich merge-conflicten voordoen bij het samenvoegen die enkel opgelost kunnen worden door communicatie tussen de integratoren en de auteur van de branch—dit kan misverstanden en tijdverlies opleveren.
- Hoe langer een branch een onafhankelijk leven leidt, hoe moeilijker de integratie zal verlopen. Intussen zijn er immers nog wijzigingen door andere teamleden geïntegreerd die impact kunnen hebben op nieuwe toevoegingen.

Pull-requests (PR's) zijn eigenlijk hetzelfde als branches die aangeboden worden om samen te voegen met de master-branch. Bij deze werkwijze is het zo dat elk teamlid op een aparte repository werkt, een zgn. fork. Zij maken dan topic-branches aan op de eigen fork waar ze wijzigingen aanbrengen. Als dit werk klaar is, kan je via GitHub deze wijzigingen als een Pull Request naar de originele repository sturen. Dit heeft als voordeel dat wijzigingen gecontroleerd kunnen geïntegreerd worden met de master-branch. Het wordt ook gebruikt in projecten waar slechts een kernteam schrijftoegang heeft op de centrale repository, wat voor ons niet van toepassing is. Werken met forks en pull-requests geeft echter nog een extra laag van complexiteit, zoals het synchroniseren van wijzigingen in de centrale repository naar alle forks. Voor dit project is het gebruik van forks en pull requests als werkwijze dan ook af te raden. Het gebruik van branches is voor een project als deze met een beperkt aantal vaste ontwikkelaars voldoende.

4.1.4. Trunk based development

Onder *trunk based development*¹ verstaan we de werkwijze waarbij er geen branches aangemaakt worden, maar elk teamlid rechtstreeks op de *trunk*, de masterbranch commit.

De eenvoudigste (maar gevaarlijke) werkwijze is:

```
$ git pull
# ... wijzigingen ...
$ git add .
$ git commit --message "Beschrijvende boodschap"
$ git push
```

¹ <https://trunkbaseddevelopment.com/>

Je haalt de laatste code binnen van de centrale Git repository, maakt lokaal wijzigingen die je registreert (bij voorkeur in verschillende kleine commits) en terug pusht. Er zijn echter enkele gevaren:

- Op het moment dat je een push uitvoert, kunnen er al andere wijzigingen gebeurd zijn. Je moet dan opnieuw een pull uitvoeren en alle wijzigingen samenvoegen (merge) met je lokale wijzigingen.
- Een merge maakt de historiek van de code complexer: het “pad” tussen commits splitst op en komt opnieuw samen. Als dit vaak gebeurt, wat in een groot team waarschijnlijk is, dan wordt het moeilijker de historiek te begrijpen.

Je kan dit vermijden door volgende werkwijze toe te passen:

```
$ git pull
# ... wijzigingen ...
$ git add .
$ git commit --message "Beschrijvende boodschap"
$ git pull --rebase
$ git push
```

Het rebase commando op lijn 5 haalt alle wijzigingen op GitHub binnen, maar zal eerst alle externe wijzigingen toepassen en pas daarna je eigen commit(s). Het zal dus lijken alsof jouw wijzigingen slechts gedaan zijn na de laatste externe commit. Er zal dus geen opsplitsing in het pad tussen commits gebeuren en de historiek blijft lineair.

Als je de globale optie “**pull.rebase**” aangezet hebt, dan is het niet nodig de optie “**--rebase**” te specificeren. Pas daarvoor de volgende aanbevolen basisinstellingen toe:

```
$ git config --global pull.rebase true
$ git config --global rebase.autoStash true
```

Wanneer bij een rebase merge-conflicten optreden, moet je die eerst oplossen door de getroffen bestanden te bewerken en de wijzigingen te committen. Gebruik `git status` om een overzicht te krijgen van bestanden met conflicten. Gebruik na de nodige

aanpassingen git add om aan te duiden dat het conflict is opgelost en tenslotte “git rebase --continue” om de rebase af te werken.

Als je over dit onderwerp leest in de vakliteratuur, zal je merken dat *trunk based development* vaak wordt afgeraden omdat het op die manier moeilijker is de kwaliteit van de master-branch te garanderen. Elke commit van elk teamlid introduceert potentieel fouten, terwijl je er wil voor zorgen dat de master-branch ten allen tijde klaar is om in productie te brengen. Nochtans is in omgevingen waar *Continuous Integration* toegepast wordt trunk based development de regel. Daar is men van mening dat de introductie van branches teveel complexiteit met zich meebrengt en teveel kans op merge-conflicten, zodat dit niet opweegt tegen de voordelen. Men gebruikt in dat geval andere strategieën om code die nog niet productiewaardig is uit te schakelen (bv. *feature flags*).

Je hebt nu verschillende werkwijzen gezien om met Git te werken. Kies er een van dat het best bij het team past en zorg ervoor dat je met het hele team dezelfde werkwijze hanteert.

4.2. KanBan

Om een zicht te hebben op het *work-in-process*, zullen we opnieuw gebruik maken van een KanBan -bord. Voor dit project zullen we gebruik maken van Jira¹. Het is mogelijk om bijkomende rapportering te bekomen in Jira met de plug-in Jira Assistant².

Als je een taak uitvoert voor het project, moet dit *altijd* gekoppeld zijn aan een KanBan-kaart. Je kan aan Kanban-kaarten opmerkingen toevoegen om resultaten van overleg bij te houden, om hulp te vragen aan teamleden of begeleiders (via zgn. @-mentions). Maak gebruik van de mogelijkheden van tags, mijlpalen, enz.

¹ <https://www.atlassian.com/nl/software/jira>

² <https://www.jiraassistant.com/>

4.3. Tijdregistratie

Elke student houdt *persoonlijk* een logboek bij, wat ook tijdregistratie inhoudt. Bij aanvang van elk contactmoment is dit volledig bijgewerkt.

We maken zoals eerder gemeld gebruik van Jira met een plug-in voor het rapporteren (zie documenten op Chamilo voor meer info). Wekelijks vul je je persoonlijk logboek aan met een overzicht van de in Jira geregistreerde werkuren. De begeleiders willen dat volgende zaken duidelijk in het oog springen:

- Het totaal aantal gepresteerde uren de afgelopen week;
- Cumulatief aantal gepresteerde uren sinds het begin van het project;
- Een overzicht van de KanBan-kaarten waaraan je gewerkt hebt met geschatte en gepresteerde uren.

We verwachten dat het aantal gepresteerde werkuren aan het einde van het project voor elke student de totale voorziene studietijd voor dit project benadert, met name ongeveer 200u. Dit omvat al het werk dat je voor het project presteert. Een niet-beperkende lijst van voorbeelden:

- Scripts of code schrijven;
- Technische documentatie, testplannen en -rapporten, enz. schrijven;
- Opzoekingswerk leveren om informatie te verzamelen over een uit te voeren taak;
- Aanwezigheid tijdens contactmomenten;
- Overleg met teamleden buiten de contactmomenten;
- Tijdsregistratie;
- ...

Je kan in je logboek links toevoegen naar commits, broncode, documentatie, rapporten, ... waar je auteur van bent.

Je moet altijd kunnen aantonen dat gepresteerde uren ook een tastbaar resultaat hebben opgeleverd. Dit geldt in het bijzonder ook voor opzoekwerk! Als je beweert dat je drie uur opzoekwerk hebt geleverd, maar geen enkele schriftelijke neerslag daarvan kan voorleggen, beschouwen we dit als niet gerealiseerd. Hou dus altijd nota's bij en registreer die in de team-repository.