

# Brochure Project 3: Systeembeheer

Academiejaar 2017-2018

**HoGent**  
BEDRIJF  
EN  
ORGANISATIE



# Voorwoord

Beste student(e),

Dit document bevat de richtlijnen voor het opleidingsonderdeel *Project 3: Systeembeheer*. 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 groot team

Er is één overkoepelende opgave die de gehele groep, bestaande uit studenten van campussen Gent én Aalst, samen zal realiseren. Halfweg het semester wordt het netwerk voor een eerste keer in productie gebracht. Hiervoor wordt een volledige dag gereserveerd. Daarna volgt een retrospectieve waar we samen bespreken wat goed ging en wat beter kan. We sluiten het project af met een finale productie-release, gepland op een *halve dag*.

Het werken in een grote groep zal een uitdaging worden. Je moet ten 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 netwerk.

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 terecht bij je begeleiders, of bij de coördinator van het project, Bert Van Vreckem ([bert.vanvreckem@hogent.be](mailto:bert.vanvreckem@hogent.be)).

Veel succes!

De begeleidende lectoren Project III: Systeembeheer:

Technische begeleiding:

Dirk Thijs

Karine Van Driessche

Bert Van Vreckem

Begeleiding Analyse:

Margot De Donder

Sebastiaan Labijn

Martine Van Audenrode

# Inhoudsopgave

<b>1</b>	<b>Organisatie</b>	<b>7</b>
1.1	Doel en plaats van de cursus in het curriculum	7
1.2	Leerdoelen en competenties	7
1.3	Reglementen van toepassing	8
1.4	Groepen	9
1.5	Aanwezigheid	9
1.6	Evaluatie	9
1.6.1	Eerste examenkans	10
1.6.2	Tweede examenkans	10
1.7	Begeleiding en ondersteuning	11
1.8	Dossier	11
<b>2</b>	<b>Verloop</b>	<b>13</b>
2.1	Week 1: kick-off	13

<b>2.2</b>	<b>Vóór Week 2</b>	<b>13</b>
<b>2.3</b>	<b>Week 2</b>	<b>13</b>
<b>2.4</b>	<b>Week 3 t/m 6</b>	<b>14</b>
<b>2.5</b>	<b>Week 7: Eerste productie-release</b>	<b>14</b>
<b>2.6</b>	<b>Week 8: retrospectieve</b>	<b>15</b>
<b>2.7</b>	<b>Week 9 t/m 12: voorbereiding finale productie-release</b>	<b>15</b>
<b>2.8</b>	<b>Week 13: finale productie-release</b>	<b>16</b>
<b>3</b>	<b>Opgave .....</b>	<b>17</b>
<b>3.1</b>	<b>Algemene requirements</b>	<b>19</b>
<b>3.2</b>	<b>Domeinen</b>	<b>19</b>
<b>3.3</b>	<b>Routers en switches</b>	<b>20</b>
<b>3.4</b>	<b>Linux servers VLAN 30 en 50</b>	<b>20</b>
<b>3.5</b>	<b>Windows Servers VLAN 300 en VLAN 500</b>	<b>22</b>
<b>3.6</b>	<b>Werkstations VLAN 20 en VLAN 200</b>	<b>25</b>
<b>3.7</b>	<b>Firewalls</b>	<b>26</b>
<b>3.8</b>	<b>Tools voor intern gebruik</b>	<b>26</b>
<b>4</b>	<b>Tools, teamwerk .....</b>	<b>27</b>
<b>4.1</b>	<b>Communicatie en bedrijfscultuur</b>	<b>27</b>
<b>4.2</b>	<b>Gebruik Github</b>	<b>28</b>
4.2.1	Algemene richtlijnen .....	28
4.2.2	Trunk based development .....	29
4.2.3	Branching & merging, Pull Requests .....	30
<b>4.3</b>	<b>Kanban</b>	<b>31</b>
<b>4.4</b>	<b>Tijdregistratie</b>	<b>31</b>

# 1. Organisatie

## 1.1 Doel en plaats van de cursus in het curriculum

Deze cursus is het sluitstuk van de leerlijn systeem- en netwerkbeheer binnen de bacheloropleiding. In het systeembeheerproject toon je aan dat je bekwaam bent om aan de slag te gaan als een (junior) systeem- of netwerkbeheerder, system engineer, enz.

Binnen het project zal je de competenties die je in andere cursussen opgedaan 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.

## 1.2 Leerdoelen en competenties

Zie ook de studiefiche<sup>1</sup>

### 1. Product

- (a) Kan verschillende componenten waaruit ict-infrastructuur bestaat (netwerk + services) met elkaar integreren zodat ze als geheel beschikbaar zijn voor de eindgebruikers
- (b) Kan zelfstandig complexe fysieke ict-infrastructuur (netwerk + services) opzetten, beveiligen en configureren
- (c) Kan problemen in complexe ict-infrastructuur (netwerk + services) opsporen en oplossen door het toepassen van een systematische en grondige aanpak

### 2. Documentatie

- (a) Kan gestructureerde documentatie opleveren onder supervisie van een onderwijsgevende
- (b) Kan een ruime voorbereiding maken en anticiperen op overleg

---

<sup>1</sup><https://bamaflexweb.hogent.be/BMFUIDetailxOLOD.aspx?a=87220&b=5&c=1>

## 3. Planning

- (a) Kan een korte-termijnplanning maken voor een complexe opdracht onder supervisie van de onderwijsgevende
- (b) Kan een middellange termijnplanning maken voor een complexe opdracht onder supervisie van de onderwijsgevende
- (c) Kan zelfstandig werken volgens planning op lange termijn en afspraken
- (d) Kan de opdrachten binnen het team visueel organiseren volgens het KanBan-concept
- (e) Kan een retrospectieve organiseren en de organisatie optimaliseren

## 4. Professioneel handelen

- (a) Kan spontaan en gericht op zoek gaan naar een unieke innovatieve oplossing
- (b) Kan streven naar perfectie en proactief het werk aanpassen rekening houdend met de noden van de klant.
- (c) Kan spontaan feedback vragen aan alle belanghebbenden, kan spontaan en proactief opbouwende feedback geven.

## 5. Teamwerk

- (a) Kan de voortgang van de ontwikkeling van zijn opdrachten aantonen d.m.v. een geavanceerd versiebeheersysteem.
- (b) Kan inschatten hoe hij een vlot en professioneel contact onderhoudt met medestudenten, onderwijsgevers en klanten en kan proactief communiceren wanneer deadlines niet gehaald worden.
- (c) Kan proactief reflecteren en stimuleert anderen onder supervisie van een onderwijsgevende.
- (d) Kan stimulerend samenwerken met anderen en actief bijdragen aan teamwork.
- (e) Het team is zelfsturend en kan conflicten oplossen onder supervisie van de onderwijsgevende.

## 1.3 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 volgende reglementen:

- Onderwijs- en Examenreglement (OER17-18), art. 44 *Naleving van de examenregeling*, in het bijzonder §5 (*Afwezigheid bij niet-periodegebonden evaluatie*) en §6 (gevolgen afwezigheid bij een evalatiemoment).
- Facultair Examenreglement (FOER):
  - Art. 36, *Aanwezigheid bij verplichte onderwijsactiviteiten en onderwijsactiviteiten met niet-periodegebonden evaluatie*
  - Art. 37, *Betrokkenheid bij groepswork*

De reglementen zijn te vinden via de Chamilo-cursus Studentenaangelegenheden<sup>2</sup>, Onderwijs- en examenregeling.

Verder is het ook belangrijk om de inhoud van de studiefiche te kennen<sup>3</sup>.

<sup>2</sup><https://chamilo.hogent.be/index.php?application=Chamilo%5CApplication%5CWebcms&go=CourseViewer&course=25048>

<sup>3</sup><https://bamaflexweb.hogent.be/BMFUIDetailxOLOD.aspx?a=87220&b=5&c=1>



## 1.4 Groepen

De opgave bestaat uit twee grote netwerken die onderling doorverbonden zijn. In het ene netwerk zijn alle diensten gebaseerd op Windows Server, in het andere op Linux. Elk deelnetwerk wordt door een groot team gerealiseerd. Tijdens de eerste les verdelen de studenten zich tot op zekere hoogte vrij over deze teams, met als beperking dat de studenten van beide campussen afzonderlijk gelijk verdeeld worden, en dat na de verdeling er max. slechts één persoon meer of minder in elk team mag zitten.

Bijvoorbeeld, stel dat op campus Gent 20 studenten deelnemen aan het project en op campus Aalst 9. Dan zal het Windows-team bestaan uit 10 studenten van campus Gent en 4 of 5 studenten van campus Aalst, en het Linux-team uit 10 studenten van campus Gent en resp. 5 of 4 van campus Aalst.

Studenten die na de eerste les nog niet toegewezen zijn aan een groep, worden door hun begeleiders ingedeeld.

## 1.5 Aanwezigheid

De beoordeling van het opleidingsonderdeel gebeurt op basis van “niet-periodegebonden evaluatie”. Dit betekent dat je dus voortdurend geëvalueerd zal worden tijdens de contactmomenten. Afwezigheden moet je dan ook altijd wettigen (OER, art. 41, §4.1-2). Indien je onmogelijk een bepaalde afspraak kan nakomen, verwittig dan tijdig je teamleden en je begeleider(s) en maak een andere afspraak. Registreer je afwezigheid en breng binnen de gestelde termijn je attest binnen op het studentensecretariaat.

Elke week ben je minstens twee uur aanwezig om enerzijds samen te werken aan het project, en anderzijds af te spreken met de begeleiders die jullie opvolgen. We verwachten dat jullie ook buiten deze contacturen verder werken aan jullie opdrachten. Dit valt onder de uren “zelfstudie” zoals vermeld in de studiefiche (nl. 51u).

Er zijn drie evaluatiemomenten met verplichte aanwezigheid: twee “productie-releases” in Week 7 en Week 13 en een retrospectieve in Week 8. Ongewettigde afwezigheid op één deze evaluatiemomenten (FOER, art. 36, §5) heeft als gevolg:

- Productie-release in Week 7 of Week 13: quoterings “nog niet bekwaam” voor de evaluatie van het eindresultaat van het te realiseren netwerk en services als geheel. Merk op dat je in dit geval sowieso niet kan slagen.
- Retrospectieve in Week 8: quoterings “nog niet bekwaam” voor de evaluatie van de retrospectieve.

## 1.6 Evaluatie

- Eerste examenkans: 100% niet-periodegebonden evaluatie
- Tweede examenkans: 100% niet-periodegebonden evaluatie (individuele opdracht)

### 1.6.1 Eerste examenkans

De evaluatie zal gebeuren aan de hand van zgn. *rubrics*<sup>4</sup>. Dit is een evaluatiesysteem waar je volgens verschillende categorieën en aan de hand van duidelijk beschreven indicatoren beoordeeld wordt op de mate waarin je de beoogde competenties behaald hebt. Er zijn verschillende gradaties: “nog niet bekwaam”, “bekwaam”, “gevorderd” en “deskundig”. Elke student moet *individueel* kunnen aantonen minstens “bekwaam” te zijn in *alle* technische beoordelingscriteria om te kunnen slagen. Dat kan aan de hand van opgeleverde code en documentatie (na te gaan via historiek van het versiebeheersysteem), via de wekelijkse opvolgingsgesprekken en op de evaluatiemomenten. Aan de hand van een rekenregel worden de beoordelingen voor alle criteria omgezet in een examencijfer. De scoringstabel met alle criteria en indicatoren, en rekenregels waarmee het examencijfer bepaald zal worden, wordt tijdig gepubliceerd.

Studenten die onvoldoende betrokkenheid (FOER, art. 37) vertonen bij het projectwerk, zullen uitgenodigd worden tot een functioneringsgesprek. Indien dit niet afdoende blijkt, krijgt de betrokken student als quoterings “afwezig” voor de eerste examenkans. Onder een gebrek aan betrokkenheid verstaan we onder andere:

- Niet consequent bijhouden van logboek en tijdregistratie;
- Structureel onvoldoende werkuren registreren;
- Niet kunnen aantonen een zinvolle 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 gewone contactmomenten, zonder het team of begeleiders te verwittigen;
- Niet nakomen van afspraken en beloftes naar de rest van het team of de begeleiders toe.

### 1.6.2 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 de opdracht is maandag 20 augustus 2018, 12:00u. Het dossier bestaat uit je Github-repository en bevat naast code en technische documentatie ook je logboek en tijdregistratie, volgens dezelfde verwachtingen als in de eerste examenkans. De toestand van je repository op het moment van de deadline wordt beschouwd als je inzending, latere wijzigingen worden genegeerd. Je komt een demonstratie geven van het resultaat van deze opdracht op het moment dat in het examenrooster is voorzien.

Wanneer je echter geen repository aangemaakt hebt, geen code toegevoegd hebt, of het dossier is onvolledig, dan krijg je als quoterings meteen “afwezig” en volgt er ook geen demonstratie. Omdat de categorieën “functioneren binnen het team” en “Retrospectieve” niet kunnen beoordeeld worden in de tweede examenkans, wordt de beoordeling van deze criteria tijdens de eerste examenkans ongewijzigd overgenomen naar de tweede examenkans.

---

<sup>4</sup>[https://en.wikipedia.org/wiki/Rubric\\_\(academic\)](https://en.wikipedia.org/wiki/Rubric_(academic))

## 1.7 Begeleiding en ondersteuning

Tijdens het semester worden jullie begeleid door de lectoren van de andere vakken binnen het keuzepakket: Computernetwerken, Enterprise Linux en Windows Server. Zij nemen de rol aan van *opdrachtgevers* en geven inhoudelijke begeleiding. Niet alle begeleiders zijn heel de tijd beschikbaar. Er zal een rooster gepubliceerd worden waar de aanwezigheid van de begeleiders op aangegeven is.

Wekelijks is er een **opvolgingsgesprek** met een retrospectieve en vooruitblik. Wat is er de afgelopen week gerealiseerd? 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?

## 1.8 Dossier

Het projectdossier is elektronisch en bestaat uit twee Github-repositories:

- Team-repository: hier werkt het gehele team in samen en bevat het eindproduct en bijhorende documentatie.
- Individueel logboek: door elk student individueel bijgehouden.

De team-repository omvat o.a.:

- De opdrachtoomschrijving;
- Algemene info en overzicht: IP adrestabel, afbeelding van het netwerk, werkverdeling (dit laatste kan via Github Issues/Projects of Kanban-bord), enz.;
- Het lastenboek voor elke deeltaak:
  - Specificaties en requirements;
  - Verantwoordelijke voor realisatie, verantwoordelijke voor testen;
  - Tijdschatting voor realisatie (in manuur);
  - Na realisatie: werkelijk tijdgebruik aanvullen
- Resultaten van de deeltaken:
  - Broncode (bv. van scripts, geautomatiseerde tests, enz);
  - PacketTracer-bestanden
  - enz.
- Technische documentatie, waaronder:
  - Functionele testplannen (test van individuele componenten);
  - Draaiboek productie-release;
  - Integratietestplannen (aantonen dat verschillende componenten samenwerken als geheel)

VM-images en gelijkaardige grote bestanden behoren niet tot het dossier. Als de technische documentatie goed gestructureerd word, dan geeft dit de mogelijkheid hier een e-book van te maken. Hiervoor bestaan verschillende tools: Github Pages<sup>5</sup>, Gitbooks<sup>6</sup>, readthedocs.io<sup>7</sup>, pandoc<sup>8</sup>,

<sup>5</sup><https://guides.github.com/features/pages/>

<sup>6</sup><https://www.gitbook.com/>

<sup>7</sup><https://readthedocs.org/>

<sup>8</sup><https://pandoc.org/>

Jekyll<sup>9</sup>, enz. Al deze tools gebruiken Markdown als invoerformaat.

Het individueel logboek wordt opgedeeld per lesweek en bevat voor elke week:

- Overzicht gerealiseerde taken met links naar relevante commits/documenten in de team-repo;
- Timesheet voor de afgelopen periode;
- Doelstellingen voor de volgende periode;
- Vragen/opmerkingen voor de begeleiders;

Elk teamlid houdt een persoonlijk logboek en tijdregistratie bij dat 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.

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 75 werkuren kunnen inbrengen, wat de totale begrote studietijd is volgens de studiefiche (contactmomenten + zelfstudie). Verdeeld over 13 weken komt dat op ongeveer 5u 45min per week. Dat omvat zowel de tijd van de contactmomenten als daarbuiten.

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.

---

<sup>9</sup><https://jekyllrb.com/>

## 2. Verloop

In dit deel bespreken we het verloop van het project over het semester heen.

### 2.1 Week 1: kick-off

- Toelichting organisatie door de lectoren
- Toelichting opgave
- Verdelen in groepen + toewijzen taken
- Opzetten werkomgeving
  - Kanbanbord (Trello of Github Projects)
  - Github: team repo + individuele repo
  - Persoonlijke timesheet (Toggl.com, rekenblad op Google Docs, enz.)

### 2.2 Vóór Week 2

- Starten met werken aan taken: info verzamelen, ...
- Persoonlijk logboek aanvullen
  - timesheet
  - gerealiseerde taken
  - vragen voor begeleiders (over opgave, technische moeilijkheden, ...)
  - planning voor volgende periode (= week)

### 2.3 Week 2

- Stel vragen aan de begeleiders over onduidelijkheden in de opgave

- Technische documentatie opstarten: Overzicht netwerk, IP-adrestabel, afbeelding netwerkstructuur, algemene structuur
- Maak afspraken over het gebruik van de "project management tools": Github issues, Kanbanbord, Git (branches? PRs?)

## 2.4 Week 3 t/m 6

- Tussen contactmomenten door:
  - individueel werken aan eigen taken;
  - logboek consequent bijhouden (realisaties, timesheet, planning, vragen);
- Tijdens contactmomenten:
  - opvolgingsgesprekken met begeleiders (ahv 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.

## 2.5 Week 7: Eerste productie-release

Op maandag 6 november wordt in de loop van een volledige dag het te realiseren netwerk opgezet. Dit gaat voor alle studenten door op campus Schoonmeersen, lokaal B.4.037/038.

We willen jullie hier expliciet de kans geven om fouten te maken zonder dat dit je zal aangerekend worden. Op dit moment zal het eindproduct dus nog niet geëvalueerd worden. Het belangrijkste hier is het proces: mate van automatisering, testen, troubleshooten, enz. en vooral uit deze ervaring lessen trekken voor de toekomst.

Verloop:

- 09:00–16:00: realisatie productie-release
- 16:00–16:45: demonstratie eindresultaat
- 16:45–17:00: opruimen leslokaal: **alle pc's op hun plaats en kabels correct aangesloten, netwerkkapapparaat in de kast**

Richtlijnen:

- Maak vooraf een draaiboek (onderdeel van technische documentatie).
  - Hoe worden individuele componenten opgezet?
  - Hoe ga je valideren dat de resulterende opstelling daadwerkelijk werkt? = Testplan integratietests
- Tijdens de release:
  - De gehele netwerkinfrastructuur wordt opgezet "from scratch". Wat vooraf mag voorbereid worden is het opzetten van een virtualisatieplatform (CentOS + KVM, Docker, Windows + Hyper-V, enz) en basis-images voor OS-installatie (= OS met laatste updates

- en een systeemgebruiker geconfigureerd voor remote management)
- Visualiseer de vooruitgang ahv burn-down-chart of kanban-bord
- Gebruik het draaiboek en integratietestplannen bij het opzetten van de te realiseren componenten
- Maak een verslag, bestaande uit:
  - \* testrapport: resultaten uitvoeren integratietests
  - \* troubleshooting-rapport(en): hoe heb je problemen opgespoord en opgelost?
- Achteraf: Aan de hand van de reflectievragen hieronder, denk je na over het verloop en resultaat. Noteer dit in je logboek:
  - Wat ging goed?
  - Wat ging minder goed? Waar heb je fouten gemaakt?
  - Wat heb je geleerd? Heb je ideeën voor verbeteringen aan je eigen werkwijze of gerealiseerde componenten? Kan de werking van het team als geheel verbeterd worden? Op welke manier kan je gemaakte fouten in de toekomst voorkomen?
  - Waar heb je nog problemen mee? Kan je bij iemand terecht om je daarbij te helpen of aan wie je vragen kan stellen (teamgenoot, begeleider)

## 2.6 Week 8: retrospectieve

In deze nabespreking over het eerste release-event krijg je feedback van de begeleiders over zowel de technische als organisatorische kant van het gebeuren. Elke student draagt bij aan de discussie aan de hand van de reflectievragen van hierboven. **Een belangrijke gedragsregel** tijdens de retrospectieve is dat niemand met de vinger gewezen mag worden (= **blameless retrospective**). Neem als team een gedeelde verantwoordelijkheid op over het verloop en resultaat van jullie realisatie. Fouten gebeuren nu eenmaal en niemand is volmaakt. Denk in plaats daarvan na over waarom bepaalde problemen zich hebben kunnen voordoen en hoe jullie dit de volgende keer gaan voorkomen. Vaak kan verdere automatisering menselijke fouten voorkomen. Soms is het gewoon een kwestie van ervaring opdoen en zal het een volgende keer sowieso vlotter verlopen. Of misschien kan je door elkaar beter/efficiënter te helpen een beter resultaat bekomen.

- Een van de begeleiders modereert de retrospectieve, of als een van de studenten deze rol wilt vervullen kan dat zeker ook
- Eén of twee studenten fungeren als verslaggever en houden notulen bij van de retrospectieve. Het verslag bevat:
  - Samenvatting goede punten die jullie zeker wensen te behouden
  - Samenvatting verbeterpunten, lessen uit gemaakte fouten
  - Voorstellen voor de verbetering. Dit kan gaan over technische zaken (bv. extra features, op te lossen "bugs", enz.) of organisatorische (bv. nieuwe afspraken gebruik Kanban-bord, Github, enz.)
- Waar dit relevant is, worden nieuwe Issues/tickets aangemaakt en geregistreerd in het Kanban-bord
- Begeleiders kunnen nieuwe taken en/of gewenste functionaliteiten toevoegen.

## 2.7 Week 9 t/m 12: voorbereiding finale productie-release

- Breng verbeteringen aan in de code van de te realiseren componenten
- Verbeter testplannen en productie-draaiboek

## 2.8 Week 13: finale productie-release

Op maandag 18 december, vanaf 13:30 wordt opnieuw het gehele netwerk “from scratch” opgezet, maar deze keer krijgen jullie daar veel minder tijd voor. De ervaring die jullie tijdens de eerste productie-release hebben opgedaan, aangevuld met de retrospectieve, en de voorbereiding tijdens de laatste weken, zou jullie in staat moeten stellen om dit te realiseren!

Na afloop beoordelen de technische begeleiders het resultaat aan de hand van een uitgebreide demonstratie.

Verloop:

- 13.30–16.30: realisatie productie-release
- 16.30–17.15: demo, eindbeoordeling product
- 17.15–17.30: opruimen leslokaal



### 3. Opgave

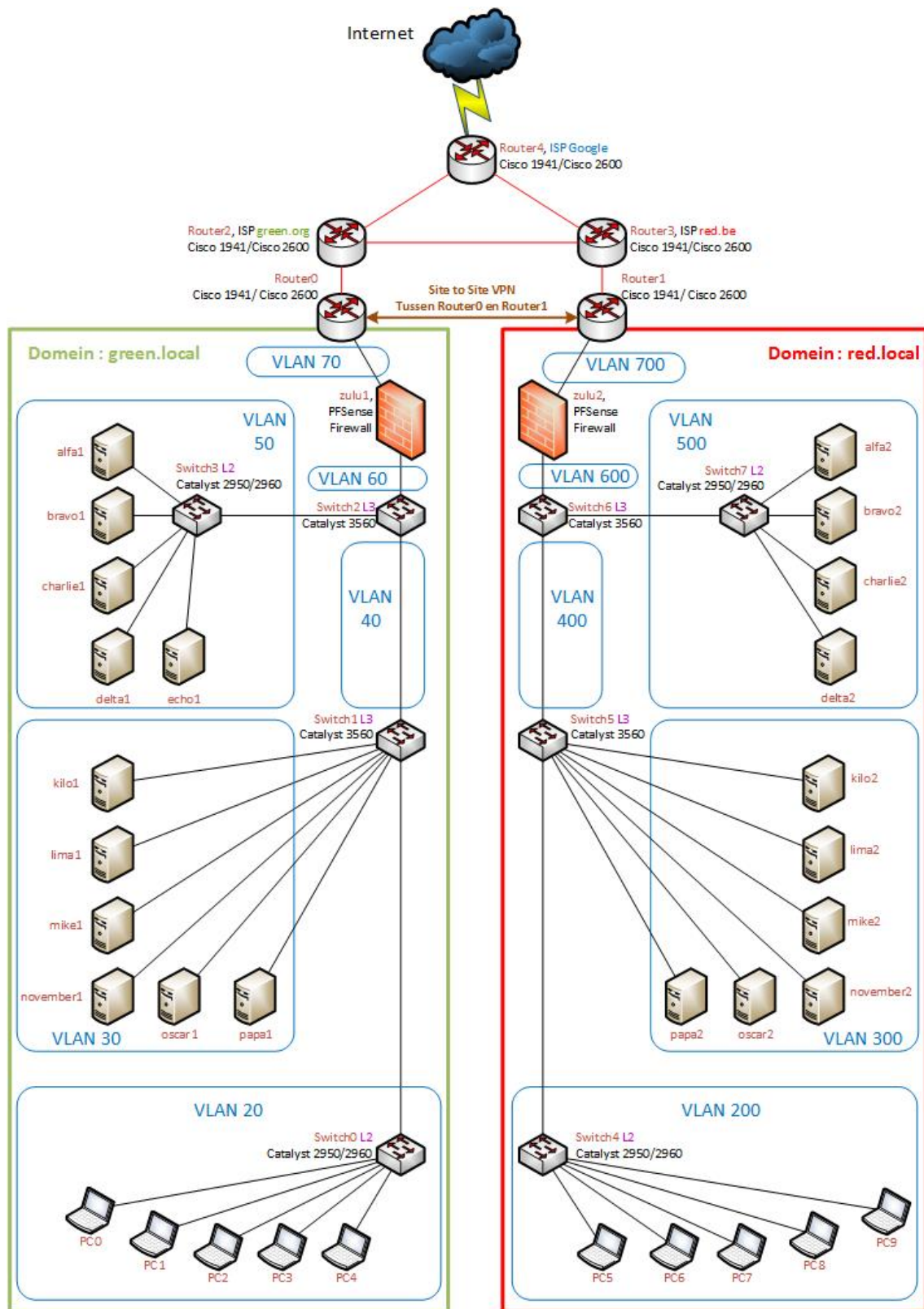
In deze opdracht is het de bedoeling om in één team twee complete netwerkdomeinen op te zetten, onderling doorverbonden, incl. alle typische diensten: DNS, web, e-mail, enz. Het is aan het team om te beslissen wie welke taken op zich zal nemen om die uit te werken, te testen en op te leveren.

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 bijgesteld 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. Spreek dus goed af en maak 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.

Een overzicht van het te realiseren netwerk is te vinden in Figuur 3.1.



Figuur 3.1: Overzicht van het te realiseren netwerk.

### 3.1 Algemene requirements

- De gehele opstelling moet volledig geautomatiseerd zijn. Dat betekent dat op het moment van een productie-release het netwerk “from scratch” kan opgezet worden.
  - In het Linux-netwerk (zie verder) wordt gebruik gemaakt van het configuration management system Ansible
  - In het Windows-netwerk wordt gewerkt met hetzij PowerShell, Desired State Configuration of Ansible.
- Op elke server is een administrator-account voorzien voor beheer vanop afstand (Linux: via ssh; Windows: via WinRM/HTTPS).
- Elke component in het netwerk stuurt informatie over de eigen werking door naar het monitoringsysteem.
- Als testomgeving kan je VirtualBox met Vagrant gebruiken, maar de productie-omgeving moet een ander platform zijn. Gebruik bv. KVM-QEMU (Linux), Hyper-V (Windows), Docker (beide), OpenNebula, ...
- Als hostsystemen voor de productie-omgeving kan gebruik gemaakt worden van de oude klaspc's (Dell Optiplex 760). Zorg dat die tijdig klaar zijn om gebruikt te kunnen worden!
- Optioneel: implementeer IPv6 binnen het eigen netwerk.

### 3.2 Domeinen

Dit zijn de domeinnamen en toegewezen IP-blokken, per team:

- *green.local*: 172.16.0.0/16 (Linux)
- *red.local*: 172.18.0.0/16 (Windows)

Het domein *green.local* is volledig op Linux gebaseerd en *red.local* is een volledig Windows domein.

Opgelet: Bovenvermelde Net-ID's moeten nog verder verdeeld worden over de verschillende VLANs m.b.v. VLSM. De routing tussen de VLANs wordt uitgevoerd met statische routing.

Elk domein heeft vier switches en 5 VLANs :

- **2 L2-switchen:** Verbinden alle computersystemen binnen een VLAN.
- **2 L3-switchen:** Zorgen voor de inter-VLAN verbindingen.
- **VLAN 20, 200:** interne clients.
  - Private, dynamische IP-adressen (via DHCP)
  - Toegang tot de publieke servers van de andere domeinen.
- **VLAN 30, 300:** interne servers.
  - Vaste IP adressen.
  - Enkel bereikbaar voor hosts van het eigen domein.
- **VLAN 40, 400:** Verbinding tussen de 2 L3 switchen.
  - Vaste IP adressen.
- **VLAN 50, 500:** publiek toegankelijke servers
  - Vaste IP adressen
  - Bereikbaar voor interne clients én voor clients van andere domeinen.
- **VLAN 60, 600:** Verbinding naar de PFSense Firewall.
- **VLAN 70, 700:** Verbinding naar het router netwerk en de buitenwereld.

### 3.3 Routers en switches

Alle routers zijn met elkaar verbonden met seriële kabels of UTP-kabels en maken gebruik van een dynamisch routeringsalgoritme. Doorverbinden met het Internet wordt voorzien via de PFSense firewall.

ACLs op de routers en/of switchen beperken het binnenkomende verkeer van andere domeinen tot de publieke servers en de protocollen die op deze servers gebruikt worden.

Test vooraf de werking van deze opstelling uit in PacketTracer. Automatiseer het opzetten van de fysieke apparatuur aan de hand van scripts, of met een configuration management system<sup>1</sup>.

*Router4*, die een verbinding vormt tussen beide netwerken en het Internet is een gedeelde verantwoordelijkheid van beide teams.

### 3.4 Linux servers VLAN 30 en 50

#### **alfa1**

een domeinserver voor green.local voor gecentraliseerd beheer van gebruikers. Er zijn twee mogelijke pistes om dit te realiseren. Enerzijds via een LDAP-server (OpenLDAP, 389 Directory Server), anderzijds via een Linux-versie van de Active Directory Domain Controller (Samba 4). De verantwoordelijke kiest zelf welk platform gebruikt wordt.

Client-pc's hebben geen eigen gebruikers, authenticatie gebeurt telkens via de domeinserver.

Maak onderstaande afdelingen (groepen) aan.

- IT Administratie
- Verkoop
- Administratie
- Ontwikkeling
- Directie

Maak een duidelijk verschil tussen gebruikers, computers en groepen. Voeg enkele gebruikers en minstens 5 werkstations toe (één in IT-Administratie, één in Ontwikkeling, één in Verkoop, één in Administratie en één in Directie).

#### **bravo1**

*Authoritative-only* DNS server voor green.local, gebaseerd op BIND. DNS-requests voor andere domeinen worden NIET beantwoord. De buitenwereld kent deze server als “*ns1*”

#### **charlie1**

Een “slave DNS” server die zelf geen zonebestanden bevat, maar synchroniseert met *bravo1*.

De buitenwereld kent deze server als “*ns2*”

<sup>1</sup>Zie bv. [https://docs.ansible.com/ansible/latest/list\\_of\\_network\\_modules.html#ios](https://docs.ansible.com/ansible/latest/list_of_network_modules.html#ios)

**delta1**

Een mailserver (gebaseerd op Postfix) met SMTP en IMAP. Het moet mogelijk zijn mails te versturen en te ontvangen tussen de twee domeinen. Gebruikers in de directory server hebben ook een mailbox. Deze mailserver past ook anti-spam en anti-virus technieken toe.

**echo1**

een publiek toegankelijke webserver met HTTP/HTTPS en voorzieningen voor dynamische webapplicaties (LAMP stack, of variant daarop) en Drupal. Het database-deel van de webapplicatie wordt uitbesteed aan *november1*. Je kan naar deze webserver surfen met het “www” voorvoegsel (vb. <https://www.green.local/>) vanop elke host in alle domeinen.

**kilo1**

Een DHCP-server voor de interne clients. Gekende werkstations krijgen een IP-adres gereserveerd op MAC-adres.

Hosts die booten over het netwerk (PXEBoot), worden doorverwezen naar de PXEBoot-server, *papa1*.

**lima1**

Een interne file-server die aanspreekbaar is met SMB/CIFS. Deze file server is geïntegreerd in de directory structuur gedefinieerd door *alfa1* en voorziet elke Linux-gebruiker van een home folder, enkel toegankelijk voor de gebruiker zelf. Wanneer een gebruiker inlogt op een workstation, is die home folder meteen zichtbaar.

**mike1**

Een content management system naar keuze voor intern gebruik. Mogelijke voorbeelden van een CMS zijn Wordpress, Drupal, Joomla, Mediawiki, ...

Het database gedeelte van deze CMS server staat eveneens op de database server *november1*.

Deze server is enkel toegankelijk voor interne gebruikers van het domein green.local.

**november1**

Een MariaDB database-server voor intern gebruik. Bevat al minstens de databases voor servers *echo1* en *mike1*. De database-admins besteden bijzondere zorg aan performantie en high availability.

**oscar1**

Een monitoringsysteem dat de toestand van het gehele netwerk + services kan visualiseren, gebaseerd op de Elastic Stack.

Afhankelijk van het type apparaat worden minstens volgende parameters opgevolgd:

- Cisco routers en switches (via SNMP):
  - CPU-gebruik
  - Geheugengebruik
  - Inkomend verkeer op alle netwerkpoorten
  - Uitgaand verkeer op alle netwerkpoorten
- Linux servers (zowel fysieke hosts als VMs):
  - CPU-gebruik
  - Geheugengebruik
  - Toestand van de harde schijf(f)(ven)
  - Toestand van de specifieke service die op de machine draait (bv. metriecken over database-queries, HTTP-requests, DNS-requests, enz.)
- Werkstations:
  - CPU-gebruik
  - Geheugengebruik
  - Toestand harde schijf(f)(ven)
  - Gebruikerslogins

De resultaten zijn raadpleegbaar via de Kibana webinterface die ook op *oscar1* draait, via url <https://mon.green.local/>.

### **papa1**

Een PXEBoot-server die toelaat om hosts over het netwerk te installeren. Voorzie via een menu twee mogelijkheden: server of workstation.

Een server wordt geïnstalleerd met de laatste versie van CentOS, en is via Kickstarter automatisch voorgeconfigureerd om in het netwerk opgenomen te worden (bv. administrator-gebruiker, package-installatie, configuratie updates).

Een workstation wordt geïnstalleerd met Fedora Workstation, dat voorgeconfigureerd werd via Kickstarter. Na installatie kan het workstation meteen gebruikt worden op het netwerk (bv. gebruikerslogin, toegang netwerkshares, aanvaarden certificaten webservers, enz.), en stuurt het informatie over de werking door naar het monitoringsysteem.

### **quebec1**

Een forwarding DNS-server voor werkstations. DNS-requests voor green.local worden geforward naar ns[12].green.local; DNS-requests voor red.be worden geforward naar ns[12].red.local; alle andere requests gaan naar een geschikte DNS-server die externe namen kan resolvable (bv. die van HoGent). Gebruik hiervoor Dnsmasq.

## **3.5 Windows Servers VLAN 300 en VLAN 500**

### **alfa2**

Domeincontroller met DNS. Dit is de Master DNS server voor het domein red.local. Externe DNS-requests worden doorgegeven aan de daarvoor meest geschikte server. De buitenwereld kent deze server als “ns1”.

Client-pc's hebben geen eigen gebruikers, authenticatie gebeurt telkens via de domeincontroller. Maak hiervoor onderstaande afdelingen (groepen) aan.

- IT Administratie
- Verkoop
- Administratie
- Ontwikkeling
- Directie

Maak een duidelijk verschil tussen gebruikers, computers en groepen. Voeg enkele gebruikers en minstens 5 werkstations toe (één in ITAdministratie, één in Ontwikkeling, één in Verkoop, één in Administratie en één in Directie). Werk met zwervende profielen (roaming profiles).

Werk volgende beleidsregels uit op gebruikersniveau:

- verbied iedereen uit afdelingen de toegang tot het control panel
- verwijder het games link menu uit het start menu voor alle afdelingen
- verbied iedereen uit de afdelingen Administratie en Verkoop de toegang tot de eigenschappen van de netwerkadapters.

Zorg voor de juiste toegangsgroepen voor de fileserver (Modify/Read/Full) en voeg de juiste personen en/of groepen toe. Meer info vind je onder *lima2*, de file-server. Maak gebruik van AGDLP (Account, Global, Domain Local, Permission) voor het uitwerken van de groepsstructuur.

## **bravo2**

Een tweede Domain controller die eveneens dienst doet als tweede DNS-server voor het domein. De buitenwereld kent deze server als “ns2”.

## **charlie2**

Een mailserver (Exchange Server) met SMTP en IMAP. Gebruik steeds de meest recente versie. Het moet mogelijk zijn mails te versturen en te ontvangen tussen de twee domeinen. Gebruikers in de directory server (AD) hebben ook een mailbox.

## **delta2**

Een publiek toegankelijke webserver met HTTP/HTTPS en voorzieningen voor dynamische webapplicaties (ASP.NET) en een demo-applicatie. Maak gebruik van een database. Het database-deel van de webapplicatie wordt uitbesteed aan *november2*. Je kan naar deze webserver surfen met het “www” voorvoegsel (vb. <https://www.red.be/>) vanop elke host in alle domeinen.

## **kilo2**

Een interne member-server in het domein red.local die dienst doet als DHCP-server voor de interne clients.

Disk	Volume	Type	File System	Volume label	Share
1	C:	Primair	NTFS	System	
	D:	Primair	NTFS	VerkoopData	X
	E:	Primair	NTFS	OntwikkelingData	X
2	F:	Primair	NTFS	ITData	X
	G:	Primair	NTFS	DirData	X
	H:	Primair	NTFS	AdminData	X
2	Y:	Primair	NTFS	HomeDirs	X
	Z:	Primair	NTFS	ProfileDirs	X

Tabel 3.1: Overzicht volumes op de file-server *lima2*

Volume	Read	Write	Full control
VerkoopData	Verkoop	Verkoop	IT Administratie
OntwikkelingData	Ontwikkeling	Ontwikkeling	IT Administratie
ITData	IT Administratie	IT Administratie	IT Administratie
DirDATA	Directie	Directie	IT Administratie
AdminData	Administratie	Administratie	IT Administratie
HomeDirs	Alle Afdelingen	Alle Afdelingen	IT Administratie
ProfileDirs	Alle Afdelingen	Alle Afdelingen	IT Administratie
ShareVerkoop	Verkoop, Ontwikkeling	Verkoop	IT Administratie

Tabel 3.2: Overzicht toegangsrechten per volume op de file-server *lima2***lima2**

Een interne file-server. Zorg voor schijven voor de verschillende afdelingen, zoals aangegeven in tabel 3.1. Pas toegangsrechten toe zoals aangegeven in tabel 3.2.

**OPGELET!** De directories HomeDirs en ProfileDirs zijn toegankelijk voor iedereen. In de HomeDirs folder zitten alle home directories van alle domeingebruikers. Uiteraard kan elke gebruiker enkel in zijn eigen home directory lezen en schrijven. Alle andere gebruikers krijgen geen toegang. Er wordt net hetzelfde gedaan met de profile directories in de ProfileDirs folder.

Stel volgende quota in:

- VerkoopData, DirData en AdminData maximum 100MB per gebruiker!
- OntwikkelingData en ITData maximum 200Mb per gebruiker!
- Stel in dat er voor AdminData dagelijks een schaduwkopie wordt gemaakt van de data.

Maak een map ShareVerkoop aan die gedeeld wordt met Verkoop en Ontwikkeling met toegangsrechten zoals in Tabel 3.2 aangegeven.

**mike2**

Een member server in het domein red.local. Deze server doet dienst als Intranet/CMS server (Sharepoint) die enkel toegankelijk is voor interne systemen uit het domein red.local.

Het database gedeelte van deze Sharepoint server staat eveneens op de database server *november2*.

Als inhoud voor deze CMS server voorzie je alle Windows documentatie van dit project.



**november2**

Een member server in het domein red.local. Deze server is een Microsoft MS SQL-Server voor *mike2* (Sharepoint) en voor *delta2* (publieke webserver).

**oscar2**

Een member server in het domein red.local. Deze server is een Real-Time Monitoring server die alle Windows Servers opvolgt. Deze Server maakt gebruik van de monitoring applicatie PRTG.

Volgende parameters worden minimaal opgevolgd :

- CPU gebruik.
- Geheugen gebruik.
- Toestand van de harde schijf(f)(ven).
- Toestand van elke specifieke service die op een Windows server staat de draaien. (services zoals MS-SQL, Sharepoint, IIS, mail, ...)

**papa2**

Een member server in het domein red.local. Deze server is een Microsoft System Center Server. Gebruik steeds de meest recente versie.

Deze server is verantwoordelijk voor het deployen van Windows 7 Enterprise en/of Windows 10 Enterprise images naar de Windows clients in VLAN 200.

Daarnaast zorgt deze server voor alle Windows en Office updates voor zowel de clients als de servers.

Deze Server moet ook in staat zijn om een aantal basis applicaties te deployen op de client PC's. Volgende applicaties moeten zeker kunnen uitgerold worden:

- Office 2013 of Office 2016
- Adobe Acrobat Reader laatste versie.
- Java Packages laatste versie.
- Adobe Flash componenten voor IE en Firefox.

## 3.6 Werkstations VLAN 20 en VLAN 200

Configureer op zijn minst 5 client pc's waar een gebruiker kan op inloggen, e-mail kan lezen, en van waar de publieke en private services van het eigen en publieke services van andere domeinen kunnen getest worden.

Elk van deze PC's is lid van één van gespecificeerde afdelingen in de directorystructuur (AD voor Windows en OpenLDAP voor Linux).

Zorg ervoor dat elke afdeling minimaal met 1 PC voorzien is.

### 3.7 Firewalls

- **zulu1** bevindt zich tussen VLANs 60 en 70.
- **zulu2** bevindt zich tussen VLANs 600 en 700.
- OS: De meest recente stabiele versie van PFSense.
- Deze Firewall heeft NAT uitgeschakeld! NAT is actief op de router *Router0*.
- Configureer deze firewall zodanig dat enkel die poorten openstaan die echt nodig zijn binnen uw netwerk.
- Configureer deze firewall zodanig dat je vanuit elk subnet van je netwerk/LAN (zowel de VLANs als de router subnets) kan communiceren met het internet.

### 3.8 Tools voor intern gebruik

Naast het opleveren van netwerkinfrastructuur naar klanten toe, bestaat een deel van het werk van een systeembeheerder ook uit het optimaliseren van het “productieproces”. Voorzie dus dat een deel van de tijd hier aan besteed wordt, en dat dit zichtbaar is in het kanban-bord.

Voorbeelden:

- Opzetten virtualisatie-infrastructuur voor de productie-omgeving;
- Automatiseren van tests, o.a.:
  - Statische analyse van code op stijl en vaak voorkomende fouten (“linting”);
  - Functionele tests op individuele componenten;
  - Integratietests;
- Uitvoeren van geautomatiseerde tests na elke commit of merge;
- Automatiseren van genereren technische documentatie in de vorm van een e-book of website (zoals [readthedocs.io](https://readthedocs.io));
- Schrijven of aanpassen van tools/scripts voor gebruik binnen het project;
- enz.

## 4. Tools, teamwerk

In dit project zullen jullie samenwerken in een groot team, wat tot nu toe nog nooit 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.

Hier 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 vlot toegankelijk is.

### 4.1 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 met een groot team dat verspreid zit over twee locaties en meestal op aparte momenten samenkomt. Een goede, open communicatie is daarbij essentieel. We zullen er naar streven dat niemand reden heeft om zich uitgesloten of onvoldoende geïnformeerd te voelen.

Enkele richtlijnen:

- Er wordt **geen gebruik gemaakt van Facebook** voor onderlinge communicatie, enkel van kanalen die geschikt zijn voor professioneel gebruik (zie verder).
- Als je mondeling afspraken maakt met teamleden, maak dan onmiddellijk een verslag dat gepubliceerd wordt voor heel het team.
- Gebruik eventueel 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.

- Goede plaatsen om resultaten van overleg te publiceren zijn o.a. de kanban-kaarten van Github Projects of Trello (zie verder) of de technische documentatie.
- Een andere mogelijkheid is om een Slack-kanaal op te zetten voor communicatie binnen het team (het is mogelijk om de werking ervan te integreren met Github).

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...

## 4.2 Gebruik Github

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.

### 4.2.1 Algemene richtlijnen

Eerst en vooral is een goede, **overzichtelijke directorystructuur** belangrijk. Merge-conflicten 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 hebt. Aan boodschappen als “wijzigingen,” “fix,” “herwerken,” “brol,” “nog brol,” enz. heeft niemand iets, ook je toekomstige zelf niet. 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 algemene aanduidingen als [doc], [fix], enz. Het nummer van de Github issue kan ook informatief zijn, bv. [bravo#25], [doc#42], enz.

**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 Github, des te meer kans op merge-conflicten. Commit dus meerdere keren per werksessie, en telkens je één concrete stap vooruit raakt. **Synchroniseer** ook minstens één keer per werksessie (aan het einde) met Github.

### 4.2.2 Trunk based development

Onder *trunk based development*<sup>1</sup> verstaan we de werkwijze waarbij er geen branches aangemaakt worden, maar elk teamlid rechtstreeks op de *trunk*, de master-branch commit.

De eenvoudigste werkwijze is:

```
1 $ git pull
2 [... wijzigingen ...]
3 $ git add .
4 $ git commit --message "Beschrijvende boodschap"
5 $ git push
```

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

- Op het moment dat je een push naar Github 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:

```
1 $ git pull
2 [... wijzigingen ...]
3 $ git add .
4 $ git commit --message "Beschrijvende boodschap"
5 $ git fetch origin
6 $ git rebase origin/master
7 $ git push
```

Het commando op lijn 5 haalt alle wijzigingen op Github binnen, *maar houdt dit bij op de achtergrond en overschrijft je lokale bestanden niet*. Het commando op lijn 6 zal je lokale historiek herschrijven en eerst alle externe wijzigingen toepassen en tenslotte 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.

Soms gebeurt het toch dat bij de rebase merge-conflicten optreden. In dat geval moet je die eerst oplossen door de getroffen bestanden te bewerken en de wijzigingen te committen.

Als je over dit onderwerp leest, 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*

---

<sup>1</sup><https://trunkbaseddevelopment.com/>

*Integration* toegepast wordt trunk based development de regel. Daar is men van mening dat de introductie van branches teveel complexiteit met zich meebrengt die niet opweegt tegen de voordelen.

### 4.2.3 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:

```
1 $ git pull
2 $ git checkout --branch feature/newstuff
3 [... wijzigingen ...]
4 $ git add .
5 $ git commit --message "Beschrijvende boodschap"
6 $ 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 github issue/ticket toe te voegen aan de branch-naam, bv. feature/22-dns-server, fix/56-dhcp-wont-start.

Na het pushen naar Github, kan dan verder gewerkt worden aan een ander ticket. De volgende stap is er dan voor zorgen dat alle wijzigingen op Github 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 commit-boodschap toe te voegen.

De voordelen van deze werkwijze zijn o.a.:

- het is makkelijker te garanderen dat de master-branch ten allen tijde een werkende versie van het project bevat.
- aan het mergen kan er een QA-proces gekoppeld worden dat introductie van fouten vermijdt (bv. geautomatiseerde testen, linter, ...).
- 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 afspraken houden.
- De verantwoordelijken voor de integratie van code 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 deze werkwijze dan ook af te raden.

## 4.3 Kanban

Om een zicht te hebben op het *work-in-process*, zullen we opnieuw gebruik maken van een kanban-bord. Het team bepaalt zelf welke tool daarvoor gebruikt wordt. Twee voor de hand liggende keuzes zijn Github Projects en Trello<sup>2</sup>.

Het voordeel van Github Projects is dat het meteen geïntegreerd is in de werking van het versiebeheersysteem en Github Issues.

Trello is ook een populaire tool hiervoor en heeft de mogelijkheid om te integreren met Github via een plugin (*Power-Up*).

Als je voor het project werkt, zou dit *altijd* moeten gekoppeld zijn aan een kanban-kaart. Gebruik de kanban-kaarten 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.

## 4.4 Tijdregistratie

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

Je kiest zelf de tool die je daarvoor het meest geschikt vindt, maar let wel op volgende zaken:

- In je wekelijks rapport moet een overzicht komen van de gepresteerde uren sinds het vorige contactmoment.
- Na afwerken van een taak, moet je het totaal gependeerde tijd aan die taak kunnen vergelijken met de vooraf gemaakte inschatting.
- Je moet altijd kunnen aantonen dat gepresteerde uren ook een tastbaar resultaat hebben opgeleverd.

Het meest eenvoudige is een rekenblad (bv. Google Docs) met bv. volgende kolommen:

---

<sup>2</sup><https://trello.com/>

- Dag
- Gepresteerde tijd in uren (rond af tot een half uur, het is weinig zinvol dit op de minuut nauwkeurig te doen)
- Taak (met verwijzing naar Github issue/kanban-kaart)
- Beschrijving (kort)

Voeg in dat geval ook een link toe vanuit je logboek naar dit rekenblad zodat je begeleiders altijd inzage hebben.

Je kan ook een (web)app voor tijdregistratie gebruiken, zoals Toggl<sup>3</sup>.

---

<sup>3</sup><https://toggl.com/>