



Maykin Media

Technisch ontwerp Werkbezoekapplicatie

Versie 1.1
29 januari 2024
WA de Landgraaf

Vertrouwelijk



Inhoudsopgave

Inleiding.....	3
Overzicht WBA applicatie.....	3
Koppelvlakken in WBA.....	4
Infrastructuur overzicht.....	5
SP diensten.....	6
Maykin diensten.....	7
Deployment van nieuwe releases.....	7
Architectuur WBA applicatie.....	8

Inleiding

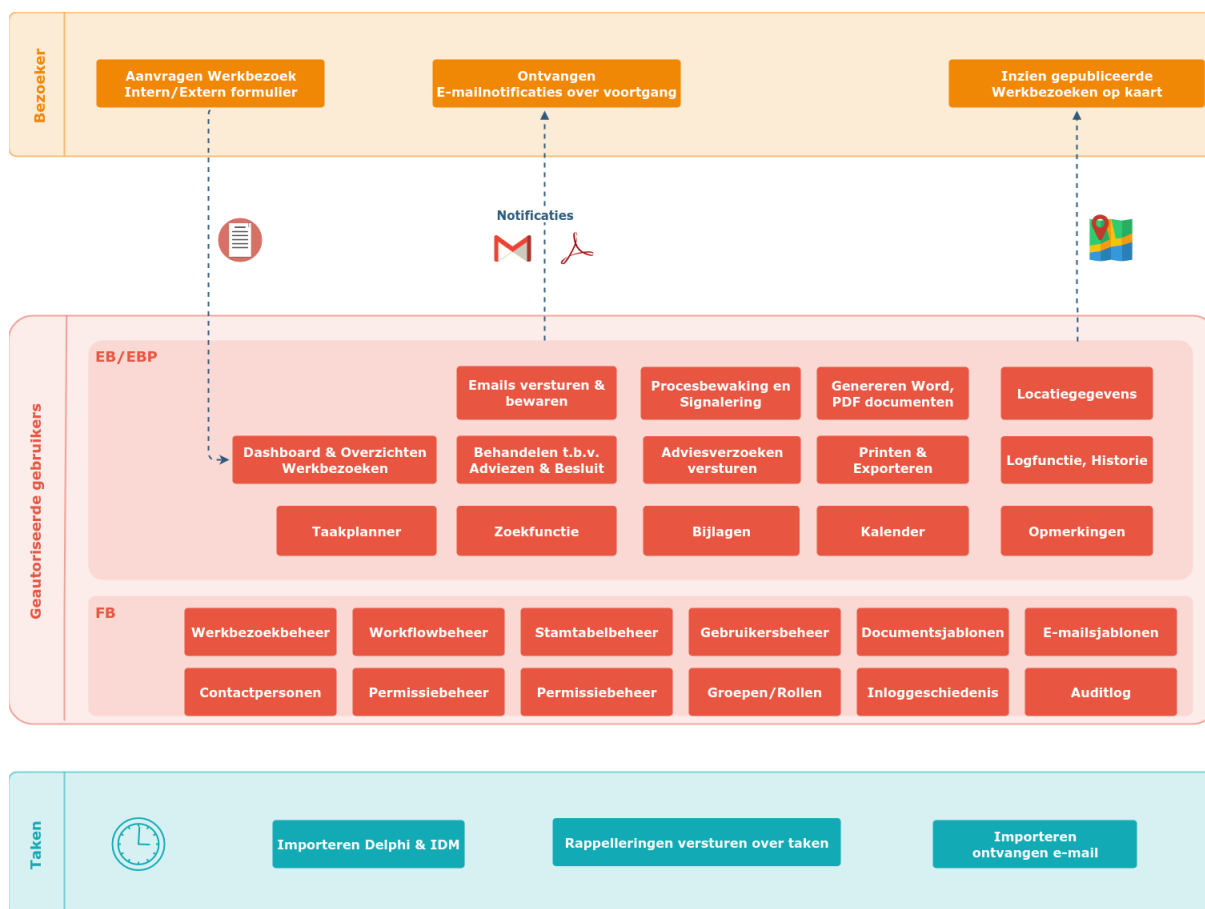
In 2020 heeft Maykin Media in opdracht van de Ministerie Infrastructuur en Waterstaat Directie Communicatie (MinlenW DCO) de nieuwe Werkbezoekapplicatie (WBA) gerealiseerd. In dit document beschrijven wij de opgeleverde oplossing in technische zin.

Overzicht WBA applicatie

De WBA applicatie is ontwikkeld om EB/EBP te ondersteunen in:

1. het eenvoudig en zo volledig mogelijk laten indienen van verzoeken tot werkbezoeken en externe optredens door bewindspersonen;
2. het op een gestructureerde wijze faciliteren van het proces van beoordelen, advies inwinnen, nemen van een besluit tot aan de daadwerkelijke bijeenkomst;
3. het beperkt toegang bieden tot werkbezoekgegevens aan verschillende geautoriseerde personen;
4. het inzichtelijk maken van ingeplande werkbezoeken en externe optredens en andere informatie in de tijd op de kalender;
5. het weergeven van uitgevoerde werkbezoeken en externe optredens op een publiekelijk inzichtelijke kaart;

Een schematische weergave van de functionaliteiten in WBA:





Koppelvlakken in WBA

Er wordt in de applicatie van de volgende externe bronnen gebruik gemaakt:

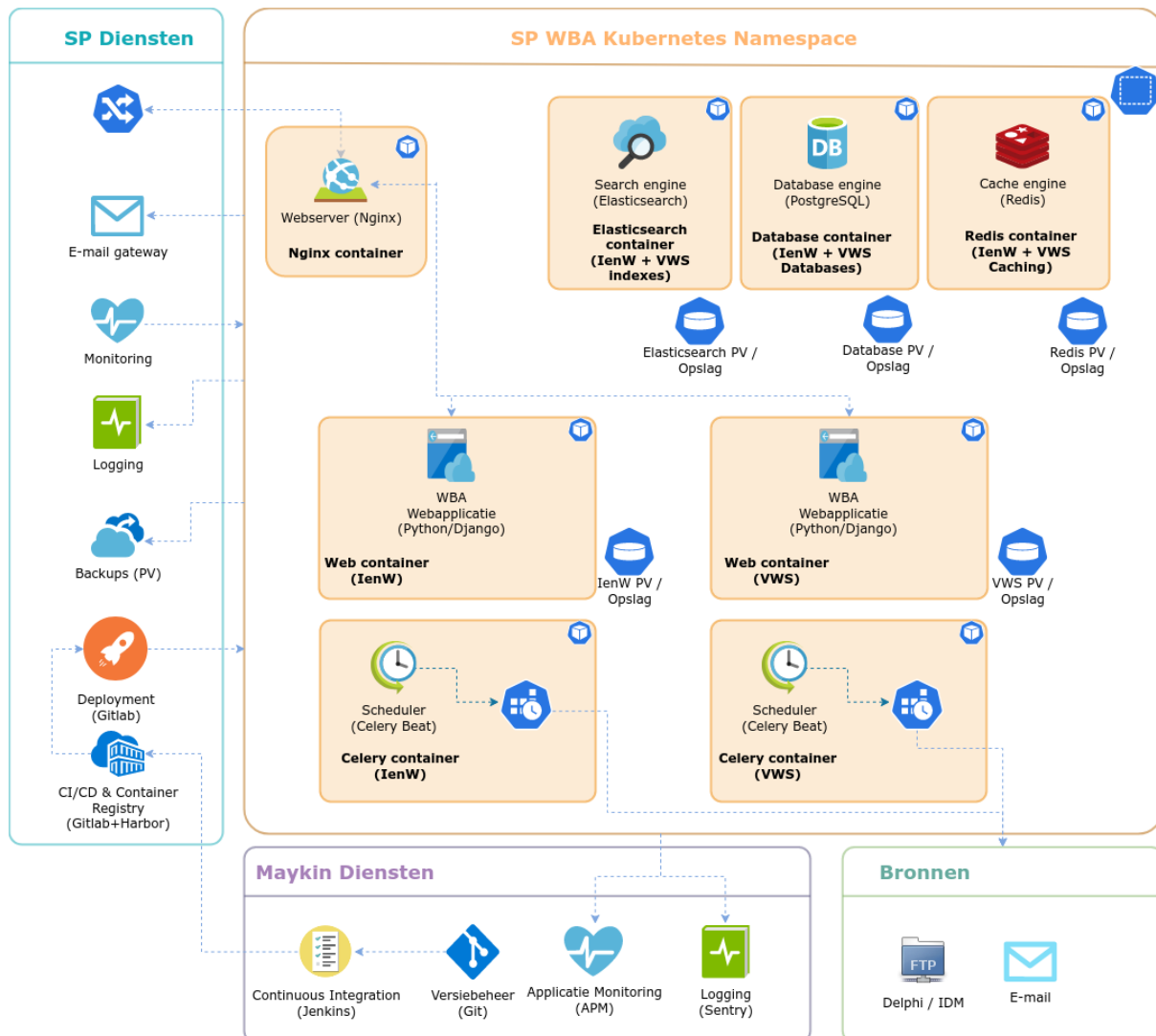
- Delphi → WBA. Algemeen Overleg betreffende de bewindspersoon, via SFTP
- IDM → WBA. Contactgegevens medewerkers (alleen MinVWS), via SFTP
- Inkomende e-mail → WBA. Automatisch inlezen en koppelen van e-mail (optioneel)

Er zijn geen API's of andere koppelvlakken vanuit WBA naar andere systemen toe.



Infrastructuur overzicht

WBA maakt gebruik van een Kubernetes cluster draaiend op het Standaard Platform (SP) van MinlenW (binnenkort: Logius). De installatie is als volgt schematisch weergegeven:



De WBA installatie op de Kubernetes cluster van het SP bestaat uit de volgende containers per acceptatie/(pre)productie namespace. Wanneer we het hebben over een WBA 'cluster' bedoelen we de containers binnen deze namespace.

- **Web:** Bevat de WBA webapplicatie, welke in Python/Django gebouwd is conform de voorgestelde architectuur. Handelt de applicatie-specifieke webverzoeken van browsers af.
- **Nginx:** De Nginx installatie binnen deze container is geconfigureerd als reverse-proxy, deze webserver is voornamelijk ingericht om media en statische bestanden rechtstreeks te serveren.
- **Database:** Een standaard PostgreSQL container is ingericht voor het gestructureerd opslaan van alle WBA data.



- Elasticsearch: Een standaard Elasticsearch container is ingericht voor het snel volledig doorzoekbaar maken van de werkbezoeken in WBA. De index in de zoekmachine wordt dagelijks ververst, en wanneer er wijzigingen zijn aan de werkbezoeken in WBA.
- Redis: Een standard Redis container is ingericht voor het snel ophalen van tijdelijke data (caching) en voor het opslaan van de Celery task queue data.
- Celery: Een kopie van WBA (web container) ingericht om de asynchrone taken uit te voeren op gezette tijden. Een voorbeeld is het regelmatig sturen van rappellingen aan gebruikers over taken of het automatisch synchroniseren van de applicatie via de externe bronnen.

Binnen deze namespace worden 2 installaties van de applicatie ondersteund: een voor MinVWS en een voor MinlenW. Dit wordt mogelijk gemaakt door het draaien van 2 instanties van de Web en Celery images. Er zijn dus 4 gedeelde containers (Nginx, Database, Elasticsearch, Redis) en 4 ministerie-specifieke containers (Web-lenW, Web-VWS, Celery-lenW, Celery-VWS). Elke ministerie heeft tevens een eigen persistent volume (PV) voor de opslag van bestanden.

Door deze opzet worden afhankelijkheden tussen beide installaties qua opslag en capaciteit beperkt, beide ministeries hebben een volwaardig eigen installatie. Waar er overlap is (de 4 standaard containers) is hiervoor gekozen vanwege de ingestelde SP resource quota i.c.m. de mogelijkheid om het gebruik van deze containers eenvoudig in de toekomst te kunnen scheiden.

Het uitgangspunt met deze inrichting is dat het opschalen met nieuwe ministeries of afsplitsen van ministeries in een eigen namespace of infrastructuur eenvoudig mogelijk moet blijven. Met deze opzet zorgen wij ervoor dat flexibiliteit, kosten en complexiteit in balans blijven zonder dat de gemaakte keuzes niet meer teruggedraaid kunnen worden.

Er is tijdens de bouw van de applicatie een kaartmodule overgenomen vanuit de voorloper van WBA, deze module is overgenomen en geïntegreerd in de applicatie. De kaartlagen zijn als static images opgeslagen op de PV van de applicatie.

Vanwege de SP / lenW-specifieke eisen is er geen gebruik gemaakt van de reguliere installatie playbooks van Maykin. De verschillende componenten zijn voor een webapplicatie zoals WBA standaard, echter de configuratie, deployment en beheer van WBA vereisen een hoge mate van afwijking t.o.v. onze standaard 'operating procedures' dat ervoor gekozen is om dit volledig los te splitsen. Er is wel gebruik gemaakt van de eerder gerealiseerde SP-specifieke Kubernetes 'manifests' voor Newsroom lenW Live waarmee we de installatie van applicaties conform de SP aansluitvoorwaarden relatief eenvoudig hebben weten te realiseren.

SP diensten

Naast de WBA cluster zijn er een reeks diensten van het Standaard Platform een afhankelijkheid is voor het goed kunnen functioneren van WBA en waar Maykin gebruik van maakt voor technisch applicatiebeheer. Uiteraard heeft het SP meer diensten in gebruik, slechts een deel zijn voor het technisch ontwerp van WBA onderkend:

- Loadbalancer / Ingress: Verkeer van en naar WBA verloopt via dit onderdeel van het SP. Handelt SSL-termination af en stuurt de verzoeken (afhankelijk van de domeinnaam) door naar een WBA cluster.



- E-mail gateway: Verkeer verzonden vanuit WBA wordt via de centrale SP E-mailgateway ienwprd1-mailrelay001.external-cloud.nl verstuurd. Er is een optionele inkomende mailverkeer
- Monitoring: Via de OpenShift en Gitlab omgevingen van het SP is inzicht mogelijk in het correct draaien van de containers.
- Logging: Via de Kibana omgeving van het SP is inzicht in de Kubernetes logs van de verschillende containers
- Backups: Snapshots van Persistent Volumes via Velero

Maykin diensten

Vanwege de beschikbaar gestelde SP diensten waar WBA op aangesloten dient te worden zijn maar een deel van de reguliere diensten van Maykin in gebruik voor WBA.

- Logging: Via een centrale Sentry-omgeving wordt loginformatie verzameld en wordt (bij warnings/errors/exceptions in WBA) het technische beheerteam van Maykin er van op de hoogte gebracht. Hierdoor valt bij problemen van een applicatie snel een analyse uit te voeren en kan Maykin inspelen op problemen zonder dat ze door een gebruiker of beheerder gerapporteerd hoeven te worden.
- APM: Applicatie-monitoring wordt ingezet voor het controleren van de beschikbaarheid en de performance van WBA. Net zoals bij het SP wordt hiervoor de ELK-stack gebruikt.

Deployment van nieuwe releases

Een nieuwe WBA release volgt de volgende stappen:

1. Bij Maykin wordt in git een pull-request van develop naar master uitgevoerd. Na het slagen van alle tests en een code-review worden de wijzigingen samengevoegd in de master-branch ter release.
2. Via Ansible wordt door Maykin op de eigen OTAP-infrastructuur de WBA test- en staging-omgevingen automatisch bijgewerkt (werkbezoek-test.maykin.nl, werkbezoek-ienw.maykin.nl, werkbezoek-vws.maykin.nl)
3. De WBA repository wordt door Maykin gepushed naar de Gitlab omgeving van het SP, inclusief de Kubernetes configuratiebestanden. De test- en acceptatie-images worden via de Gitlab Build Pipeline gebouwd en in de Gitlab/Harbour registry geupload.
4. Op basis van de images in de registry wordt de Kubernetes cluster opgezet conform de in WBA ingerichte Kubernetes configuratie.

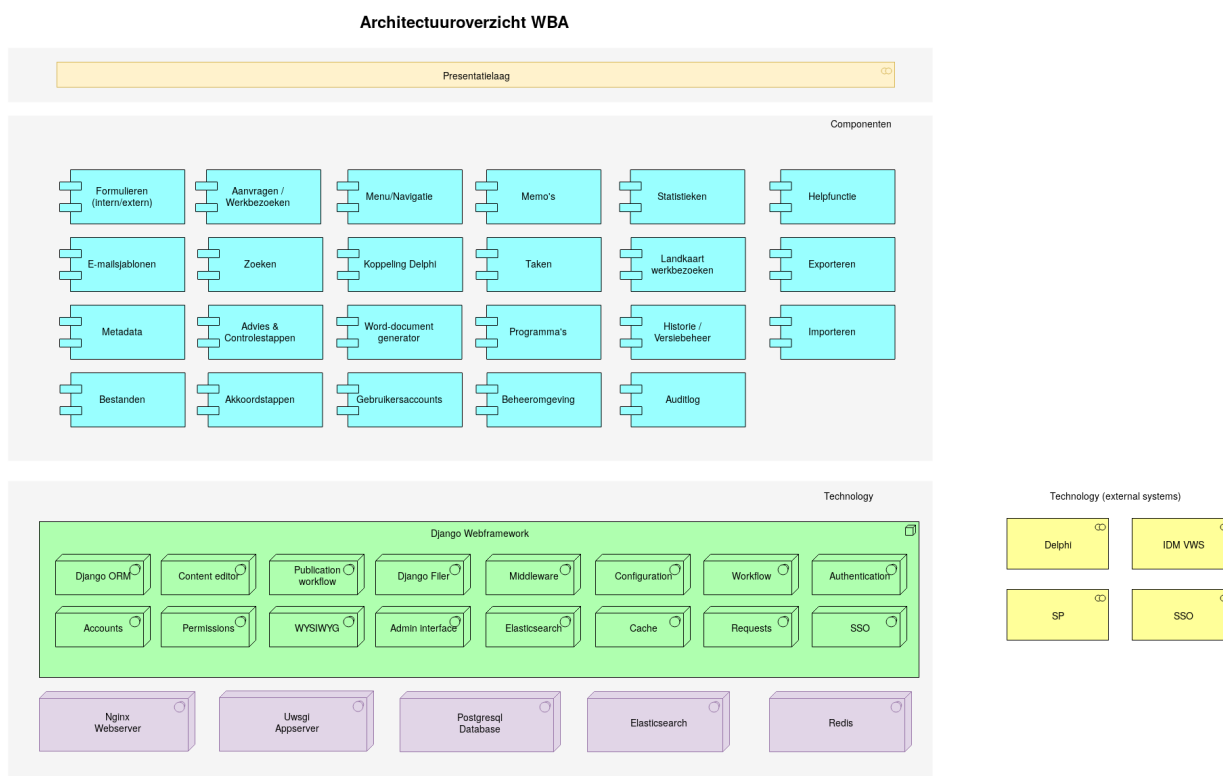


Architectuur WBA applicatie

Wij hebben voor WBA omgeving deels gebruik maken van standaard (open source) componenten, gecombineerd met modules die wij voor vergelijkbare projecten hebben gerealiseerd. Daar bovenop is maatwerk gerealiseerd om de door DCO gewenste workflow mogelijk te maken.

Voor de onderliggende componenten zijn gebruik gemaakt voor de (voor Maykin) standaard onderdelen: PostgreSQL als database, Nginx als webserver, Elasticsearch als search engine en Redis als caching server. Deze en de overige gebruikte componenten van haar oplossingen vallen onder een open source licentie, door onze “open source tenzij” werkwijze sluiten wij hier in goed aan op de werkwijze van het SP en wordt “vendor lock-in” voorkomen.

In onderstaand architectuurschets worden de verschillende modules en componenten weergegeven. Deze modulaire aanpak zorgt ervoor dat bij doorontwikkeling het toevoegen van nieuwe modules relatief eenvoudig is.



Het webframework Django wat wij gebruiken voor de realisatie van vrijwel al ons projecten is modulair van aard. Door nieuwe modules toe te voegen kunnen bestaande website functionaliteiten worden uitgebreid of nieuwe onderdelen worden toegevoegd. Mochten modules niet meer nodig zijn dan zijn deze ook eenvoudig weer uit te schakelen.