Document Generator - Complete Implementatie Handleiding

Auteur: Manus AI
Versie: 1.0
Datum: December 2024

Inhoudsopgave

Inleiding

De Document Generator is een complete standalone webapplicatie ontwikkeld voor Google Cloud Platform die gebruikersinput omzet naar professionele PDF documenten met behulp van Google Docs sjablonen. Deze uitgebreide handleiding biedt stap-voor-stap instructies voor de implementatie, configuratie en het onderhoud van het complete systeem.

Projectoverzicht

Het Document Generator systeem is ontworpen als een modulaire, schaalbare oplossing die organisaties in staat stelt om automatisch documenten te genereren zoals offertes,

facturen, werkbonnen en gecombineerde facturen. Het systeem integreert naadloos met Google Workspace en biedt een moderne webinterface voor gebruikersbeheer, klantbeheer, productbeheer en documentgeneratie.

De applicatie bestaat uit drie hoofdcomponenten: een React-gebaseerde frontend voor gebruikersinteractie, een Flask-gebaseerde backend API voor bedrijfslogica en gegevensverwerking, en een PostgreSQL database voor gegevensopslag met Google Sheets synchronisatie voor eenvoudig beheer. Het systeem maakt gebruik van Google Docs API voor sjabloonverwerking en Google Drive API voor documentopslag.

Kernfunctionaliteiten

Het systeem biedt een uitgebreide set functionaliteiten die zijn ontworpen om de volledige documentgeneratie workflow te ondersteunen. De gebruikersinterface is gebouwd met moderne webstandaarden en biedt een responsief dashboard dat zich aanpast aan verschillende gebruikersrollen binnen een organisatie. Monteurs zien andere informatie dan verkopers, en administrators hebben toegang tot uitgebreide configuratiemogelijkheden.

De documentgeneratie functionaliteit ondersteunt dynamische placeholder vervanging met loop-ondersteuning voor herhalende elementen zoals productregels. Het systeem kan automatisch documentnummers genereren, BTW berekeningen uitvoeren, en totalen berekenen. Alle gegenereerde documenten worden opgeslagen in Google Drive met gestructureerde mappenstructuur voor eenvoudige organisatie.

Het klant- en productbeheer systeem biedt volledige CRUD functionaliteit met zoek- en filtermogelijkheden. Gebruikers kunnen tijdens het documentgeneratie proces nieuwe klanten en producten toevoegen zonder de workflow te onderbreken. Het systeem ondersteunt ook projectgebaseerde organisatie waarbij documenten gekoppeld kunnen worden aan specifieke projecten of opdrachten.

Technische Specificaties

De applicatie is gebouwd met moderne technologieën en best practices voor enterprise-grade applicaties. De frontend gebruikt React 18 met TypeScript ondersteuning, Tailwind CSS voor styling, en Vite als build tool voor optimale performance. De backend is ontwikkeld in Python met Flask framework, SQLAlchemy ORM voor database interacties, en JWT voor authenticatie.

Voor deployment wordt gebruik gemaakt van Docker containers die draaien op Google Cloud Run voor automatische schaling en hoge beschikbaarheid. De database draait op Google Cloud SQL met automatische backups en point-in-time recovery. Het systeem

gebruikt Google Secret Manager voor veilige opslag van gevoelige configuratie zoals API keys en database credentials.

De CI/CD pipeline is geïmplementeerd met GitHub Actions en ondersteunt automatische testing, security scanning, en deployment naar staging en productie omgevingen. Het systeem is ontworpen voor multi-tenant gebruik waarbij meerdere organisaties veilig gebruik kunnen maken van dezelfde applicatie instantie.

Architectuur Overzicht

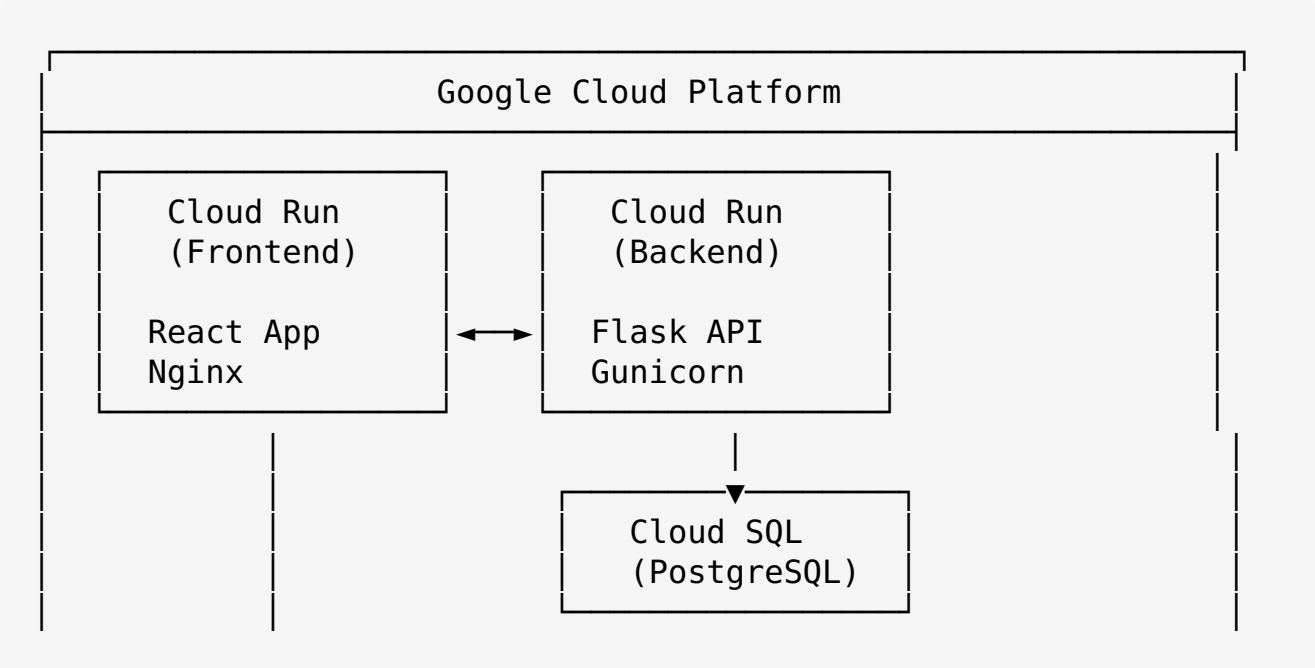
Systeemarchitectuur

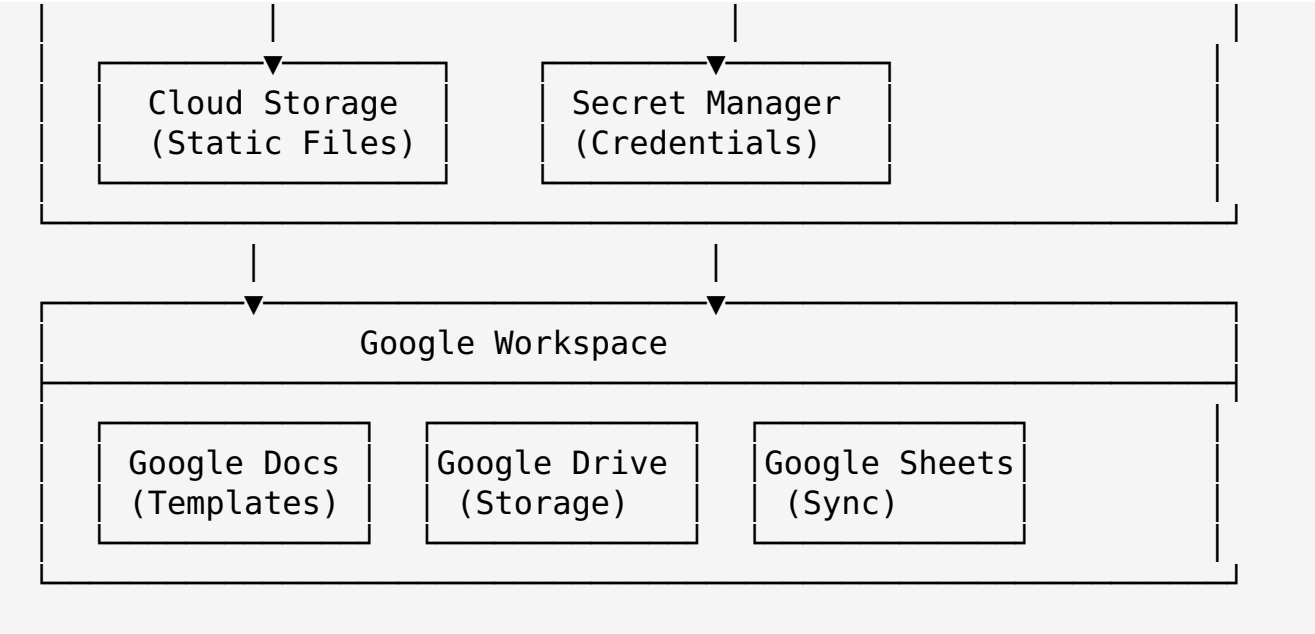
De Document Generator applicatie volgt een moderne microservices architectuur die is geoptimaliseerd voor cloud-native deployment op Google Cloud Platform. Het systeem is ontworpen met scheiding van verantwoordelijkheden, waarbij elke component een specifieke rol heeft in de totale applicatie workflow.

De architectuur bestaat uit vier hoofdlagen: de presentatielaag (React frontend), de applicatielaag (Flask backend API), de gegevenslaag (PostgreSQL database), en de integratielaag (Google Workspace services). Deze gelaagde benadering zorgt voor maximale flexibiliteit, onderhoudbaarheid en schaalbaarheid.

De frontend applicatie communiceert uitsluitend via RESTful API endpoints met de backend, wat zorgt voor een duidelijke scheiding tussen gebruikersinterface en bedrijfslogica. Dit ontwerp maakt het mogelijk om in de toekomst eenvoudig mobile applicaties of andere clients toe te voegen zonder wijzigingen aan de backend.

Component Diagram





Data Flow Architecture

De gegevensstroom in het systeem volgt een gestructureerd patroon dat zorgt voor consistentie en betrouwbaarheid. Wanneer een gebruiker een document wil genereren, wordt de request eerst gevalideerd door de frontend voordat deze wordt doorgestuurd naar de backend API. De backend haalt de benodigde gegevens op uit de database, verwerkt de sjabloon via Google Docs API, en slaat het resulterende document op in Google Drive.

De database synchronisatie met Google Sheets gebeurt bidirectioneel via een geautomatiseerd proces dat wijzigingen in beide richtingen synchroniseert. Dit zorgt ervoor dat gebruikers zowel via de webapplicatie als via Google Sheets gegevens kunnen beheren, waarbij consistentie gegarandeerd blijft.

Voor authenticatie en autorisatie gebruikt het systeem JWT tokens die worden uitgegeven na succesvolle login. Deze tokens bevatten gebruikersinformatie en rechten, waardoor de backend kan bepalen welke acties een gebruiker mag uitvoeren. Het systeem ondersteunt ook OAuth2 integratie met Google accounts voor naadloze single sign-on ervaring.

Security Architecture

De beveiligingsarchitectuur is ontworpen volgens industry best practices en compliance vereisten. Alle communicatie tussen componenten gebeurt via HTTPS met TLS 1.3 encryptie. API endpoints zijn beveiligd met JWT authenticatie en role-based access control (RBAC) voor granulaire autorisatie.

Gevoelige gegevens zoals database credentials, API keys, en encryptie sleutels worden opgeslagen in Google Secret Manager met automatische rotatie. Het systeem

implementeert defense-in-depth principes met meerdere beveiligingslagen op netwerk-, applicatie-, en data niveau.

Audit logging wordt geïmplementeerd voor alle kritieke acties zoals document generatie, gebruikersbeheer, en configuratiewijzigingen. Deze logs worden centraal opgeslagen en kunnen worden gebruikt voor compliance rapportage en security monitoring.

Scalability Design

Het systeem is ontworpen voor horizontale schaling waarbij individuele componenten onafhankelijk kunnen worden geschaald op basis van vraag. Cloud Run services schalen automatisch van nul tot duizenden instanties op basis van inkomende requests. De database kan worden geschaald via read replicas en connection pooling.

Voor high-availability scenarios kan het systeem worden gedeployed in meerdere regio's met load balancing en failover mechanismen. De stateless architectuur van de backend maakt het mogelijk om instanties dynamisch toe te voegen of te verwijderen zonder impact op gebruikers.

Caching strategieën zijn geïmplementeerd op meerdere niveaus, inclusief browser caching voor statische assets, application-level caching voor database queries, en CDN caching voor globale content distributie.

Vereisten en Voorbereiding

Technische Vereisten

Voor een succesvolle implementatie van het Document Generator systeem zijn verschillende technische vereisten van toepassing. Deze vereisten zijn onderverdeeld in ontwikkelomgeving, productieomgeving, en externe services om een duidelijk overzicht te bieden van wat nodig is voor elke fase van het project.

Ontwikkelomgeving Vereisten

De ontwikkelomgeving vereist een moderne setup die ondersteuning biedt voor containerized development en cloud-native workflows. Een werkstation met minimaal 8GB RAM en 50GB vrije schijfruimte is aanbevolen voor optimale performance tijdens development en testing. Het besturingssysteem kan Windows 10/11, macOS 10.15+, of een moderne Linux distributie zijn.

Voor de development tools is Node.js versie 18 of hoger vereist voor de frontend development, samen met pnpm als package manager voor optimale dependency

management. Python 3.11 of hoger is nodig voor backend development, bij voorkeur geïnstalleerd via pyenv voor versie management. Docker Desktop is essentieel voor lokale containerized testing en moet worden geconfigureerd met minimaal 4GB geheugen toewijzing.

Git versie 2.30 of hoger is vereist voor version control, bij voorkeur geconfigureerd met SSH keys voor secure repository access. Een moderne code editor zoals Visual Studio Code met relevante extensions voor Python, JavaScript, en Docker development wordt sterk aanbevolen voor productiviteit.

Google Cloud Platform Vereisten

Een actief Google Cloud Platform account met billing enabled is fundamenteel voor het project. Het account moet toegang hebben tot de volgende services: Cloud Run voor container hosting, Cloud SQL voor database services, Secret Manager voor credential management, Cloud Build voor CI/CD, en IAM voor access management.

De gebruiker moet Owner of Editor rechten hebben op het Google Cloud project om alle benodigde resources te kunnen aanmaken en configureren. Voor productie deployments wordt aanbevolen om een dedicated project aan te maken specifiek voor de Document Generator applicatie om resource isolatie en cost management te optimaliseren.

Een Google Workspace account is vereist voor de Google Docs en Drive integratie. Dit account moet administrator rechten hebben om service accounts te kunnen autoriseren en API access te configureren. De Workspace moet de Google Docs API, Google Drive API, en Google Sheets API enabled hebben.

Database Vereisten

Voor lokale development kan SQLite worden gebruikt, maar voor staging en productie omgevingen is PostgreSQL 13 of hoger vereist. De database instance moet minimaal 2GB RAM en 20GB storage hebben voor een standaard implementatie. Voor high-traffic scenarios wordt 4GB RAM en SSD storage aanbevolen.

Database backup en recovery procedures moeten worden geconfigureerd volgens organisatie policies. Google Cloud SQL biedt automatische backups en point-in-time recovery, maar custom backup strategieën kunnen worden geïmplementeerd voor specifieke compliance vereisten.

Software Dependencies

Backend Dependencies

De backend applicatie heeft verschillende Python packages nodig die zijn gespecificeerd in het requirements.txt bestand. De belangrijkste dependencies zijn Flask voor het web framework, SQLAlchemy voor database ORM, Alembic voor database migrations, en PyJWT voor authentication. Google client libraries zijn vereist voor integratie met Google Workspace services.

Voor productie deployment zijn aanvullende packages nodig zoals Gunicorn voor WSGI serving, psycopg2 voor PostgreSQL connectivity, en redis voor caching. Security packages zoals cryptography en bcrypt zijn essentieel voor password hashing en data encryption.

Frontend Dependencies

De frontend applicatie gebruikt React 18 als basis framework met TypeScript voor type safety. UI components zijn gebouwd met Tailwind CSS en shadcn/ui voor consistent design. Routing wordt afgehandeld door React Router, en state management gebruikt React Context API met custom hooks.

Build tools omvatten Vite voor development server en bundling, ESLint voor code quality, en Prettier voor code formatting. Testing dependencies zoals Vitest en React Testing Library zijn geïncludeerd voor unit en integration testing.

Account Setup en Permissions

Google Cloud Account Configuratie

De eerste stap in de account setup is het aanmaken van een nieuw Google Cloud project of het selecteren van een bestaand project voor de Document Generator implementatie. Het project moet een unieke project ID hebben die wordt gebruikt in alle configuratie bestanden en deployment scripts.

Billing moet worden enabled voor het project om gebruik te kunnen maken van betaalde services zoals Cloud Run en Cloud SQL. Het is aanbevolen om budget alerts in te stellen om onverwachte kosten te voorkomen tijdens development en testing fases.

Service accounts moeten worden aangemaakt met de juiste permissions voor verschillende componenten van het systeem. De hoofdservice account heeft toegang nodig tot Cloud SQL, Secret Manager, en Google Workspace APIs. Separate service accounts kunnen worden gebruikt voor verschillende omgevingen (development, staging, production) voor betere security isolation.

Google Workspace Configuratie

In de Google Workspace admin console moeten de benodigde APIs worden enabled: Google Docs API, Google Drive API, Google Sheets API, en Google Apps Script API. Deze APIs moeten worden geconfigureerd met de juiste OAuth scopes en domain-wide delegation indien nodig.

Service account credentials moeten worden geautoriseerd in de Workspace admin console om namens gebruikers acties uit te kunnen voeren. Dit vereist het toevoegen van de service account client ID aan de domain-wide delegation settings met de benodigde scopes.

GitHub Repository Setup

Een GitHub repository moet worden aangemaakt voor version control en CI/CD. De repository moet worden geconfigureerd met branch protection rules voor de main en production branches om accidental pushes te voorkomen. Required status checks moeten worden ingesteld voor automated testing en security scanning.

GitHub Secrets moeten worden geconfigureerd voor deployment credentials, inclusief Google Cloud service account keys, project IDs, en andere gevoelige configuratie. Deze secrets worden gebruikt door GitHub Actions workflows voor automated deployment.

Netwerk en Security Configuratie

Firewall en Network Security

Google Cloud VPC firewall rules moeten worden geconfigureerd om alleen noodzakelijk verkeer toe te staan. Cloud Run services zijn standaard publiek toegankelijk, maar kunnen worden beperkt tot specifieke IP ranges indien gewenst. Database toegang moet worden beperkt tot alleen de applicatie services.

SSL/TLS certificaten moeten worden geconfigureerd voor alle publieke endpoints. Google Cloud Load Balancer kan automatisch SSL certificaten beheren via Google-managed certificates of Let's Encrypt integratie.

Identity and Access Management

IAM policies moeten worden geconfigureerd volgens het principe van least privilege, waarbij elke service account alleen de minimaal benodigde permissions heeft. Custom IAM roles kunnen worden aangemaakt voor specifieke use cases die niet worden gedekt door predefined roles.

Multi-factor authentication moet worden enabled voor alle admin accounts die toegang hebben tot Google Cloud console en GitHub repository. Service account keys moeten worden geroteerd volgens security best practices, bij voorkeur geautomatiseerd via Secret Manager.

Pre-deployment Checklist

Voordat de daadwerkelijke deployment kan beginnen, moet een uitgebreide checklist worden doorlopen om ervoor te zorgen dat alle vereisten zijn vervuld en alle configuraties correct zijn ingesteld.

Verificatie van Google Cloud project setup omvat het controleren van enabled APIs, service account permissions, billing configuration, en resource quotas. Database connectivity moet worden getest vanuit de development omgeving om ervoor te zorgen dat alle network routes correct zijn geconfigureerd.

Google Workspace integratie moet worden gevalideerd door test API calls uit te voeren naar Google Docs, Drive, en Sheets services. Service account delegation moet worden getest om ervoor te zorgen dat de applicatie namens gebruikers documenten kan aanmaken en bewerken.

GitHub repository configuratie moet worden gecontroleerd, inclusief branch protection, secrets configuration, en workflow permissions. Lokale development environment moet volledig functioneel zijn met alle dependencies geïnstalleerd en configuratie bestanden correct ingesteld.

Google Cloud Setup

Initiële Project Configuratie

De Google Cloud setup begint met het aanmaken van een nieuw project of het configureren van een bestaand project voor de Document Generator applicatie. Deze stap is cruciaal omdat alle verdere configuratie en resources gekoppeld zijn aan dit specifieke project.

Project Aanmaken

Log in op de Google Cloud Console via console.cloud.google.com en navigeer naar de project selector in de top navigation bar. Klik op "New Project" en voer een beschrijvende naam in zoals "Document Generator Production" of "DocGen-Prod". De project ID wordt automatisch gegenereerd maar kan worden aangepast naar een meer herkenbare identifier.

Selecteer de juiste billing account voor het project. Voor productie implementaties wordt aanbevolen om een dedicated billing account te gebruiken voor betere cost tracking en budget management. Indien geen billing account beschikbaar is, moet deze eerst worden aangemaakt via de Billing sectie in de Cloud Console.

Na het aanmaken van het project, noteer de project ID zorgvuldig omdat deze wordt gebruikt in alle configuratie bestanden en deployment scripts. De project ID is immutable en kan niet worden gewijzigd na aanmaak, dus zorg ervoor dat deze correct en herkenbaar is.

API Services Activeren

De volgende stap is het activeren van alle benodigde Google Cloud APIs voor het project. Dit kan worden gedaan via de Cloud Console UI of via de gcloud CLI voor geautomatiseerde setup. De volgende APIs moeten worden geactiveerd:

Cloud Run API is essentieel voor het hosten van de containerized applicaties. Deze service biedt serverless container hosting met automatische schaling en pay-per-use pricing. Cloud Build API is nodig voor het bouwen van Docker images vanuit source code en voor CI/CD pipeline functionaliteit.

Cloud SQL Admin API moet worden geactiveerd voor database management functionaliteiten. Deze API biedt programmatic access tot database instances, gebruikers, en configuratie. Secret Manager API is cruciaal voor het veilig opslaan en ophalen van gevoelige configuratie zoals database passwords en API keys.

Google Workspace APIs moeten ook worden geactiveerd: Google Docs API voor sjabloon verwerking, Google Drive API voor document opslag, en Google Sheets API voor data synchronisatie. Deze APIs vereisen aanvullende OAuth configuratie die later in het proces wordt behandeld.

Service Account Setup

Service accounts zijn essentieel voor secure machine-to-machine communicatie tussen verschillende componenten van het systeem. Een hoofdservice account moet worden aangemaakt met de naam "document-generator" en een beschrijvende display name.

De service account moet de volgende IAM roles krijgen toegewezen: Cloud SQL Client voor database toegang, Secret Manager Secret Accessor voor het ophalen van secrets, en Storage Object Viewer voor toegang tot Cloud Storage buckets. Aanvullende custom roles kunnen worden aangemaakt voor meer granulaire access control.

Een service account key moet worden gegenereerd en veilig opgeslagen. Deze key wordt gebruikt voor authenticatie vanuit de applicatie naar Google Cloud services. Het is

belangrijk om deze key te behandelen als een wachtwoord en nooit te committen naar version control systemen.

Database Infrastructure Setup

Cloud SQL Instance Configuratie

De database setup begint met het aanmaken van een Cloud SQL instance voor PostgreSQL. Navigeer naar de Cloud SQL sectie in de Google Cloud Console en klik op "Create Instance". Selecteer PostgreSQL als database engine en kies versie 15 voor optimale performance en feature support.

Voor de instance configuratie, selecteer een machine type gebaseerd op verwachte load. Voor development en testing is db-f1-micro voldoende, maar voor productie wordt db-n1-standard-2 of hoger aanbevolen. De storage configuratie moet SSD storage gebruiken met automatische storage increase enabled om performance te optimaliseren.

Configureer de instance voor high availability indien vereist voor productie workloads. Dit creëert een standby replica in een andere zone voor automatische failover. Backup configuratie moet worden ingesteld met dagelijkse backups en een retention periode van minimaal 7 dagen.

Network configuratie is cruciaal voor security. Disable public IP indien mogelijk en gebruik private IP met VPC peering. Indien public IP noodzakelijk is, configureer authorized networks om toegang te beperken tot alleen de benodigde IP ranges.

Database en Gebruiker Setup

Na het aanmaken van de Cloud SQL instance, moet een database worden aangemaakt specifiek voor de Document Generator applicatie. Gebruik de Cloud Console of gcloud CLI om een database aan te maken met de naam "document_generator".

Een dedicated database gebruiker moet worden aangemaakt voor de applicatie met beperkte privileges. Deze gebruiker moet alleen toegang hebben tot de document_generator database en mag geen admin privileges hebben. Het wachtwoord voor deze gebruiker moet worden gegenereerd met hoge entropy en opgeslagen in Secret Manager.

Database connection strings moeten worden geconfigureerd voor verschillende omgevingen. Voor Cloud Run deployment wordt de Unix socket connection gebruikt via /cloudsql/PROJECT_ID:REGION:INSTANCE_NAME. Voor lokale development kan een TCP connection worden gebruikt met Cloud SQL Proxy.

Secret Management Configuratie

Secret Manager Setup

Google Secret Manager wordt gebruikt voor het veilig opslaan van alle gevoelige configuratie. Secrets moeten worden aangemaakt voor database credentials, JWT signing keys, en Google API credentials. Elke secret moet een beschrijvende naam hebben en worden geconfigureerd met automatische replication.

Database secrets omvatten het database wachtwoord en de volledige connection string. JWT secrets moeten worden gegenereerd met cryptographically secure random generators en minimaal 256 bits entropy hebben. Google API credentials moeten worden opgeslagen als JSON formatted secrets.

Voor multi-environment deployments moeten separate secrets worden aangemaakt voor development, staging, en production omgevingen. Dit zorgt voor proper isolation en voorkomt accidental cross-environment access.

Access Control voor Secrets

IAM policies voor Secret Manager moeten worden geconfigureerd om alleen de benodigde service accounts toegang te geven tot specifieke secrets. Het principe van least privilege moet worden toegepast waarbij elke service account alleen toegang heeft tot de secrets die nodig zijn voor zijn functie.

Secret versioning moet worden gebruikt om secret rotation te ondersteunen. Oude versies van secrets moeten worden bewaard voor rollback scenarios maar moeten worden disabled na succesvolle deployment van nieuwe versies.

Networking en Security Setup

VPC en Firewall Configuratie

Een dedicated VPC netwerk kan worden aangemaakt voor de Document Generator applicatie om network isolation te bieden. Dit is vooral belangrijk voor enterprise deployments waar network segmentatie vereist is voor compliance.

Firewall rules moeten worden geconfigureerd om alleen noodzakelijk verkeer toe te staan. Cloud Run services zijn standaard publiek toegankelijk via HTTPS, maar kunnen worden beperkt tot specifieke IP ranges indien gewenst. Database toegang moet worden beperkt tot alleen de applicatie services.

Voor enhanced security kan Cloud NAT worden geconfigureerd om outbound internet toegang te bieden voor private resources zonder public IP adressen. Dit is vooral relevant voor database instances die geen public IP hebben.

SSL/TLS Certificaat Management

SSL certificaten moeten worden geconfigureerd voor alle publieke endpoints. Google Cloud Load Balancer kan automatisch SSL certificaten beheren via Google-managed certificates. Voor custom domains moeten DNS records worden geconfigureerd om domain ownership te verifiëren.

Certificate transparency logging moet worden enabled voor alle certificaten om compliance te ondersteunen. Automatic certificate renewal moet worden geconfigureerd om certificate expiration issues te voorkomen.

Monitoring en Logging Setup

Cloud Monitoring Configuratie

Google Cloud Monitoring moet worden geconfigureerd om system health en performance te monitoren. Custom dashboards moeten worden aangemaakt voor application-specific metrics zoals document generation rate, API response times, en error rates.

Alerting policies moeten worden geconfigureerd voor kritieke metrics zoals high error rates, database connection failures, en resource utilization thresholds. Notification channels moeten worden ingesteld voor email, SMS, of Slack integratie.

Audit Logging

Cloud Audit Logs moeten worden enabled voor alle services om compliance en security monitoring te ondersteunen. Admin activity logs, data access logs, en system event logs moeten worden geconfigureerd met appropriate retention periods.

Log aggregation kan worden geconfigureerd om logs te exporteren naar BigQuery voor advanced analytics of naar externe SIEM systemen voor security monitoring.

Automated Setup Script

Voor geautomatiseerde setup is een comprehensive script beschikbaar dat alle bovenstaande configuraties uitvoert. Dit script kan worden aangepast voor specifieke requirements en omgevingen.

```
#!/bin/bash
# Voer het setup script uit
./setup-gcp.sh
```

Het script controleert prerequisites, maakt alle benodigde resources aan, configureert IAM permissions, en valideert de setup. Gedetailleerde logging wordt geboden om troubleshooting te ondersteunen indien issues optreden tijdens de setup.

Na succesvolle uitvoering van het script worden alle benodigde configuratie bestanden gegenereerd, inclusief environment files en GitHub secrets templates. Deze bestanden moeten worden gereviewed en aangepast indien nodig voor specifieke deployment requirements.

Database Configuratie

Database Schema Ontwerp

Het database schema voor de Document Generator applicatie is ontworpen volgens moderne database design principes met focus op normalisatie, performance, en schaalbaarheid. Het schema ondersteunt multi-tenant architectuur waarbij meerdere organisaties veilig gebruik kunnen maken van dezelfde database instance.

Core Entiteiten

De database bestaat uit verschillende core entiteiten die de basis vormen van het systeem. De Organizations tabel bevat informatie over bedrijven die gebruik maken van het systeem, inclusief bedrijfsnaam, contactgegevens, en configuratie instellingen. Elke organisatie heeft een unieke UUID als primary key om privacy en security te waarborgen.

De Users tabel slaat gebruikersinformatie op inclusief authenticatie credentials, persoonlijke gegevens, en role assignments. Gebruikers zijn gekoppeld aan een organisatie via een foreign key relatie. Password hashing wordt uitgevoerd met bcrypt voor optimale security, en JWT tokens worden gebruikt voor session management.

Customers tabel bevat klantinformatie per organisatie, inclusief bedrijfsgegevens, contactpersonen, en adresinformatie. De tabel ondersteunt zowel zakelijke als particuliere klanten met flexibele velden voor verschillende klanttypen. Soft delete functionaliteit is geïmplementeerd om data integriteit te behouden.

Products tabel slaat productinformatie op inclusief omschrijvingen, prijzen, BTW percentages, en categorieën. Het systeem ondersteunt zowel fysieke producten als

diensten met flexibele pricing modellen. Product varianten kunnen worden gemodelleerd via parent-child relaties.

Transactionele Entiteiten

Orders tabel bevat opdracht informatie inclusief klant referenties, status tracking, en metadata. Orders kunnen meerdere order items bevatten via een one-to-many relatie. Status transitions worden gelogd voor audit trail functionaliteit.

Order_items tabel bevat de individuele regels van een order inclusief product referenties, hoeveelheden, prijzen, en berekende totalen. Deze tabel ondersteunt complexe pricing scenarios inclusief kortingen, toeslagen, en BTW berekeningen.

Documents tabel slaat metadata op van gegenereerde documenten inclusief sjabloon referenties, Google Docs IDs, PDF URLs, en generatie timestamps. Document versioning wordt ondersteund om wijzigingen bij te houden.

Configuratie Entiteiten

Settings tabel bevat systeem en organisatie specifieke configuratie met typed values voor verschillende data types. Deze tabel ondersteunt hierarchical configuration waarbij system-wide defaults kunnen worden overschreven op organisatie niveau.

Widget_configurations tabel slaat dashboard widget configuraties op per gebruikersrol en organisatie. Dit ondersteunt het modulaire dashboard systeem waarbij verschillende gebruikers verschillende widgets kunnen zien.

Audit_logs tabel bevat een complete audit trail van alle belangrijke acties in het systeem. Deze tabel is geoptimaliseerd voor write performance en ondersteunt compliance requirements voor data retention.

Database Migraties

Migration Strategy

Database migraties worden beheerd via Alembic, het migration tool voor SQLAlchemy. Alle schema wijzigingen worden geversioned en kunnen forward en backward worden uitgevoerd voor safe deployment scenarios.

Migration bestanden worden automatisch gegenereerd op basis van model wijzigingen maar moeten handmatig worden gereviewed voor correctheid. Complex migrations die data transformaties vereisen moeten custom migration scripts gebruiken.

Voor productie deployments moeten migrations worden getest in een staging omgeving die een kopie is van de productie data. Zero-downtime migrations zijn mogelijk voor de meeste schema wijzigingen door gebruik te maken van online schema change technieken.

Initial Schema Setup

Het initiële database schema wordt aangemaakt via het database_schema.sql bestand dat alle tabellen, indexen, en constraints definieert. Dit bestand kan worden uitgevoerd op een lege database om de complete schema structuur aan te maken.

```
 -- Voorbeeld van tabel definitie
CREATE TABLE organizations (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    phone VARCHAR(50),
    address_street VARCHAR(255),
    address_postal_code VARCHAR(20),
    address_city VARCHAR(100),
    address_country VARCHAR(100) DEFAULT 'Nederland',
    created_at TIMESTAMP WITH TIME ZONE DEFAULT
CURRENT_TIMESTAMP,
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT
CURRENT_TIMESTAMP
);
```

Foreign key constraints worden gebruikt om referential integrity te waarborgen tussen gerelateerde tabellen. Cascade delete rules zijn zorgvuldig geconfigureerd om data consistency te behouden zonder ongewenste data verlies.

Data Seeding

Initial data seeding wordt uitgevoerd via het import_csv_data.py script dat voorbeelddata importeert uit CSV bestanden. Dit script kan worden aangepast om organisatie-specifieke data te importeren tijdens de initiële setup.

Het seeding proces omvat het aanmaken van een default organisatie, admin gebruiker, product categorieën, en basis configuratie instellingen. Alle seeded data gebruikt realistische maar fictieve informatie om privacy te waarborgen.

Performance Optimalisatie

Indexing Strategy

Database indexen zijn strategisch geplaatst om query performance te optimaliseren voor de meest voorkomende access patterns. Composite indexen worden gebruikt voor queries die filteren op meerdere kolommen.

Primary key indexen gebruiken UUID values voor betere distribution en om sequential key issues te vermijden. Foreign key kolommen hebben automatisch indexen voor join performance.

Text search indexen worden gebruikt voor full-text search functionaliteit op product omschrijvingen en klant namen. PostgreSQL's built-in text search capabilities worden gebruikt voor Nederlandse taal ondersteuning.

Query Optimalisatie

Database queries zijn geoptimaliseerd om N+1 query problemen te vermijden door gebruik van eager loading en join strategies. SQLAlchemy's relationship loading strategies worden gebruikt om query efficiency te maximaliseren.

Connection pooling wordt geconfigureerd om database connection overhead te minimaliseren. Pool size en timeout settings zijn afgestemd op verwachte concurrent user load.

Query caching wordt geïmplementeerd voor read-heavy operations zoals product catalogs en klant lijsten. Cache invalidation strategies zorgen voor data consistency bij updates.

Backup en Recovery

Backup Strategy

Automatische backups worden geconfigureerd via Google Cloud SQL met dagelijkse full backups en continuous transaction log backups voor point-in-time recovery. Backup retention wordt ingesteld op minimaal 30 dagen voor compliance requirements.

Cross-region backup replication kan worden geconfigureerd voor disaster recovery scenarios. Backup encryption wordt gebruikt om data privacy te waarborgen tijdens storage en transport.

Recovery Procedures

Point-in-time recovery procedures zijn gedocumenteerd voor verschillende failure scenarios. Recovery testing wordt regelmatig uitgevoerd om de effectiviteit van backup procedures te valideren.

Database restore procedures omvatten zowel full database restore als selective table restore voor specifieke data recovery scenarios. Recovery time objectives (RTO) en recovery point objectives (RPO) zijn gedefinieerd op basis van business requirements.

Google Sheets Synchronisatie

Bidirectionele Sync

Het google_sheets_sync.py script implementeert bidirectionele synchronisatie tussen de PostgreSQL database en Google Sheets. Deze synchronisatie zorgt ervoor dat wijzigingen in beide systemen worden gesynchroniseerd zonder data conflicts.

Conflict resolution strategies zijn geïmplementeerd om te bepalen welke wijziging prioriteit heeft bij simultane updates. Timestamp-based conflict resolution wordt gebruikt met manual override mogelijkheden.

Sync Monitoring

Synchronisatie logs worden bijgehouden om sync status en eventuele errors te monitoren. Automated alerting wordt geconfigureerd om administrators te informeren bij sync failures.

Data validation wordt uitgevoerd tijdens sync operations om data integrity te waarborgen. Invalid data wordt gelogd en gerapporteerd voor manual review.

Database Security

Access Control

Database toegang wordt beperkt tot alleen de applicatie service accounts met minimale benodigde privileges. Database gebruikers hebben geen admin rechten en kunnen alleen toegang krijgen tot de application database.

Connection encryption wordt afgedwongen voor alle database verbindingen. SSL certificates worden gebruikt om man-in-the-middle attacks te voorkomen.

Data Encryption

Sensitive data zoals persoonlijke informatie wordt encrypted at rest via Google Cloud SQL's transparent data encryption. Application-level encryption wordt gebruikt voor extra gevoelige velden zoals payment information.

Encryption keys worden beheerd via Google Cloud KMS met automatische key rotation. Key access wordt gelogd voor audit purposes.

Audit en Compliance

Database audit logging wordt geconfigureerd om alle data access en modifications te loggen. Audit logs worden bewaard volgens compliance requirements en kunnen worden geëxporteerd voor external audit systems.

GDPR compliance wordt ondersteund via data anonymization en deletion capabilities. Personal data kan worden geïdentificeerd en verwijderd op verzoek van data subjects.

Deployment Procedures

Pre-deployment Voorbereiding

Code Repository Setup

De deployment procedure begint met het correct configureren van de GitHub repository. Clone de repository naar uw lokale development omgeving en zorg ervoor dat alle benodigde bestanden aanwezig zijn. Controleer of de .env.example bestanden correct zijn geconfigureerd voor zowel backend als frontend componenten.

Maak een nieuwe branch aan voor deployment configuratie wijzigingen indien nodig. Dit zorgt ervoor dat deployment specifieke configuraties kunnen worden getest zonder impact op de main development branch. Branch protection rules moeten worden geconfigureerd om direct pushes naar production branches te voorkomen.

Configureer GitHub Secrets volgens de template die wordt gegenereerd door het setup script. Deze secrets bevatten gevoelige informatie zoals service account keys en database credentials die nodig zijn voor automated deployment via GitHub Actions.

Environment Configuratie

Maak kopieën van de .env.example bestanden en hernoem deze naar .env voor lokale development. Vul alle benodigde waarden in op basis van uw Google Cloud project

configuratie. Let vooral op de database connection strings, Google Cloud project ID, en API credentials.

Voor productie deployment moeten environment variabelen worden geconfigureerd via Google Secret Manager in plaats van .env bestanden. Dit zorgt voor betere security en maakt het mogelijk om configuratie te wijzigen zonder code deployments.

Valideer alle environment configuraties door de applicatie lokaal te starten en basis functionaliteit te testen. Controleer database connectivity, Google API integratie, en frontend-backend communicatie.

Lokale Development Setup

Backend Setup

Navigeer naar de document-generator-backend directory en maak een Python virtual environment aan. Activeer de virtual environment en installeer alle dependencies via pip install -r requirements.txt. Dit zorgt ervoor dat alle benodigde Python packages beschikbaar zijn.

Configureer de database connection voor lokale development. Voor eenvoudige setup kan SQLite worden gebruikt, maar voor volledige functionaliteit wordt PostgreSQL aanbevolen. Voer database migraties uit om de schema structuur aan te maken.

Start de Flask development server via python src/main.py. De server moet starten op poort 5000 en alle API endpoints moeten toegankelijk zijn. Test basis functionaliteit zoals health checks en authentication endpoints.

Frontend Setup

Navigeer naar de document-generator-frontend directory en installeer dependencies via pnpm install. Dit installeert alle benodigde Node.js packages voor de React applicatie.

Configureer de API base URL in de environment configuratie om te verwijzen naar de lokale backend server. Start de development server via pnpm run dev. De frontend moet toegankelijk zijn op poort 5173.

Test de frontend functionaliteit door te navigeren naar verschillende pagina's en te controleren of API calls correct worden uitgevoerd. Login functionaliteit en dashboard weergave moeten correct werken.

Automated Deployment via Scripts

Google Cloud Deployment Script

Het deploy-to-gcp.sh script automatiseert de volledige deployment naar Google Cloud Platform. Dit script voert alle benodigde stappen uit inclusief resource aanmaak, container builds, en service deployment.

Voordat het script wordt uitgevoerd, moet de Google Cloud CLI worden geïnstalleerd en geconfigureerd met de juiste project credentials. Voer gcloud auth login uit om te authenticeren en gcloud config set project PROJECT_ID om het juiste project te selecteren.

Het deployment script voert de volgende stappen uit: API services activeren, service accounts aanmaken, Cloud SQL instance setup, secret management configuratie, container image builds, en Cloud Run service deployment. Elke stap wordt gevalideerd voordat verder wordt gegaan.

```
# Voer deployment uit
chmod +x deploy-to-gcp.sh
./deploy-to-gcp.sh
```

Het script genereert output met URLs voor de deployed services en configuratie informatie die nodig is voor verdere setup stappen. Bewaar deze informatie zorgvuldig voor troubleshooting en onderhoud.

Database Migratie en Seeding

Na succesvolle deployment van de applicatie services moeten database migraties worden uitgevoerd om de schema structuur aan te maken. Dit gebeurt automatisch via Cloud Run jobs die worden aangemaakt door het deployment script.

Initial data seeding kan worden uitgevoerd via het import_csv_data.py script dat voorbeelddata importeert. Dit script kan worden aangepast om organisatie-specifieke data te importeren tijdens de initiële setup.

Valideer database setup door te controleren of alle tabellen correct zijn aangemaakt en dat basis data aanwezig is. Test database connectivity vanuit de applicatie door API calls uit te voeren die database toegang vereisen.

CI/CD Pipeline Setup

GitHub Actions Configuratie

De GitHub Actions workflow in .github/workflows/deploy.yml implementeert een complete CI/CD pipeline met automated testing, security scanning, en deployment naar staging en productie omgevingen.

De workflow wordt getriggerd bij pushes naar main branch (staging deployment) en production branch (productie deployment). Pull requests triggeren alleen testing en security scanning zonder deployment.

Configureer GitHub repository secrets volgens de template die wordt gegenereerd tijdens Google Cloud setup. Deze secrets worden gebruikt door de workflow voor authenticatie en deployment configuratie.

Multi-Environment Deployment

De CI/CD pipeline ondersteunt deployment naar meerdere omgevingen met verschillende configuraties. Staging omgeving wordt gebruikt voor testing en validatie voordat wijzigingen naar productie gaan.

Environment specific configuratie wordt beheerd via GitHub Environments met approval requirements voor productie deployments. Dit zorgt voor additional oversight bij kritieke deployments.

Database migraties worden automatisch uitgevoerd als onderdeel van de deployment pipeline. Rollback procedures zijn geïmplementeerd voor het geval dat migraties falen of onverwachte problemen veroorzaken.

Manual Deployment Procedures

Container Build en Push

Voor manual deployment kunnen Docker containers handmatig worden gebouwd en gepushed naar Google Container Registry. Navigeer naar elke component directory en voer docker build commando's uit.

```
 # Backend container build
cd document-generator-backend
docker build -t gcr.io/PROJECT_ID/document-generator-backend .
docker push gcr.io/PROJECT_ID/document-generator-backend

# Frontend container build
cd document-generator-frontend
```

```
docker build -t gcr.io/PROJECT_ID/document-generator-frontend .
docker push gcr.io/PROJECT_ID/document-generator-frontend
```

Zorg ervoor dat Docker is geconfigureerd voor Google Cloud authentication via gcloud auth configure-docker voordat containers worden gepushed.

Cloud Run Service Deployment

Deploy services naar Cloud Run via gcloud CLI commando's. Configureer environment variabelen, resource limits, en scaling parameters volgens productie requirements.

```
# Deploy backend service
gcloud run deploy document-generator-backend \
  --image gcr.io/PROJECT_ID/document-generator-backend \
  --platform managed \
  --region europe-west4 \
  --allow-unauthenticated

# Deploy frontend service
gcloud run deploy document-generator-frontend \
  --image gcr.io/PROJECT_ID/document-generator-frontend \
  --platform managed \
  --region europe-west4 \
  --allow-unauthenticated
```

Test deployed services door health check endpoints aan te roepen en basis functionaliteit te valideren.

Post-Deployment Configuratie

Google Workspace Integratie

Na succesvolle deployment moet Google Workspace integratie worden geconfigureerd. Dit omvat het autoriseren van service accounts voor toegang tot Google Docs, Drive, en Sheets APIs.

Upload service account credentials naar Secret Manager en configureer OAuth scopes voor domain-wide delegation. Test API toegang door document generatie functionaliteit uit te voeren.

DNS en SSL Configuratie

Configureer custom domain names voor de deployed services indien gewenst. Google Cloud Load Balancer kan worden gebruikt voor advanced routing en SSL termination.

SSL certificaten worden automatisch beheerd door Google Cloud maar kunnen worden geconfigureerd voor custom domains via DNS validation.

Monitoring en Alerting Setup

Configureer Cloud Monitoring dashboards voor application metrics en system health monitoring. Setup alerting policies voor kritieke metrics zoals error rates en response times.

Enable audit logging voor compliance en security monitoring. Configure log retention policies volgens organisatie requirements.

Rollback Procedures

Service Rollback

Cloud Run services ondersteunen traffic splitting en gradual rollouts. In geval van problemen kan traffic worden teruggerold naar een vorige versie zonder downtime.

```
# Rollback naar vorige versie
gcloud run services update-traffic document-generator-backend \
  --to-revisions=PREVIOUS_REVISION=100
```

Database Rollback

Database rollbacks zijn complexer en vereisen zorgvuldige planning. Point-in-time recovery kan worden gebruikt voor complete database restore, maar dit resulteert in data verlies.

Voor schema rollbacks kunnen reverse migrations worden uitgevoerd indien beschikbaar. Test rollback procedures in staging omgeving voordat uitvoering in productie.

Deployment Validatie

Functional Testing

Na deployment moet uitgebreide functional testing worden uitgevoerd om te valideren dat alle functionaliteiten correct werken. Test user authentication, document generatie, database operations, en Google Workspace integratie.

Voer end-to-end tests uit die de complete user workflow simuleren van login tot document generatie en download. Valideer dat alle API endpoints correct reageren en dat error handling werkt zoals verwacht.

Performance Testing

Voer performance tests uit om te valideren dat het systeem voldoet aan performance requirements. Test response times, concurrent user handling, en resource utilization onder verschillende load scenarios.

Monitor system metrics tijdens testing om bottlenecks te identificeren en resource allocation te optimaliseren indien nodig.

Security Validation

Voer security scans uit om te valideren dat alle security controls correct zijn geïmplementeerd. Test authentication en authorization mechanisms, input validation, en data encryption.

Valideer dat alle secrets correct zijn geconfigureerd en dat geen gevoelige informatie wordt geëxposeerd via logs of error messages.

Referenties

Technische Documentatie

[1] Google Cloud Run Documentation - https://cloud.google.com/run/docs
[2] Google Cloud SQL for PostgreSQL - https://cloud.google.com/sql/docs/postgres
[3] Google Secret Manager Documentation - https://cloud.google.com/secret-manager/docs
[4] Google Docs API Reference - https://developers.google.com/docs/api
[5] Google Drive API Reference - https://developers.google.com/drive/api
[6] Google Sheets API Reference - https://developers.google.com/sheets/api
[7] Flask Documentation - https://flask.palletsprojects.com/
[8] React Documentation - https://react.dev/
[9] SQLAlchemy Documentation - https://docs.sqlalchemy.org/
[10] Docker Documentation - https://docs.docker.com/

Security en Compliance

[11] Google Cloud Security Best Practices - https://cloud.google.com/security/best-practices
[12] OWASP Security Guidelines - https://owasp.org/www-project-top-ten/
[13] JWT Security Best Practices - https://tools.ietf.org/html/rfc8725
[14] PostgreSQL Security Documentation - https://www.postgresql.org/docs/current/

security.html
[15] GDPR Compliance Guidelines - https://gdpr.eu/

Development Tools

[16] GitHub Actions Documentation - https://docs.github.com/en/actions
[17] Terraform Google Cloud Provider - https://registry.terraform.io/providers/hashicorp/google/latest/docs
[18] Vite Documentation - https://vitejs.dev/
[19] Tailwind CSS Documentation - https://tailwindcss.com/docs
[20] pnpm Documentation - https://pnpm.io/

Document Versie: 1.0
Laatste Update: December 2024
Auteur: Manus AI
Contact: Voor vragen over deze implementatie handleiding, raadpleeg de technische documentatie of neem contact op met uw systeembeheerder.

Deze handleiding bevat alle benodigde informatie voor een succesvolle implementatie van het Document Generator systeem. Volg alle stappen zorgvuldig en test grondig in een staging omgeving voordat deployment naar productie.