STATENS IT

Servicespecifikation

GovCloud

Juli 2022

2022

STATENS IT

Servicespecifikation

GovCloud

8. Juli 2022

Indhold

1 Formål og indhold

Dette dokument beskriver de kundevendte komponenter, som indgår i GovCloud-platformen. Dokumentet beskriver:

- GovCloud-arkitekturen
- De komponenter som udgør GovClouds eksekveringsplatform
- De komponenter som indgår i GovClouds toolchain

Dokumentet retter sig særligt til projektledere, lead-udviklere m.fl. som skal udvikle løsninger til GovCloud-platformen, fx i forbindelse med valg af udviklingsmetoder, udviklingsværktøjer og udviklingskompetencer i en projektplanlægningsfase.

2 GovCloud

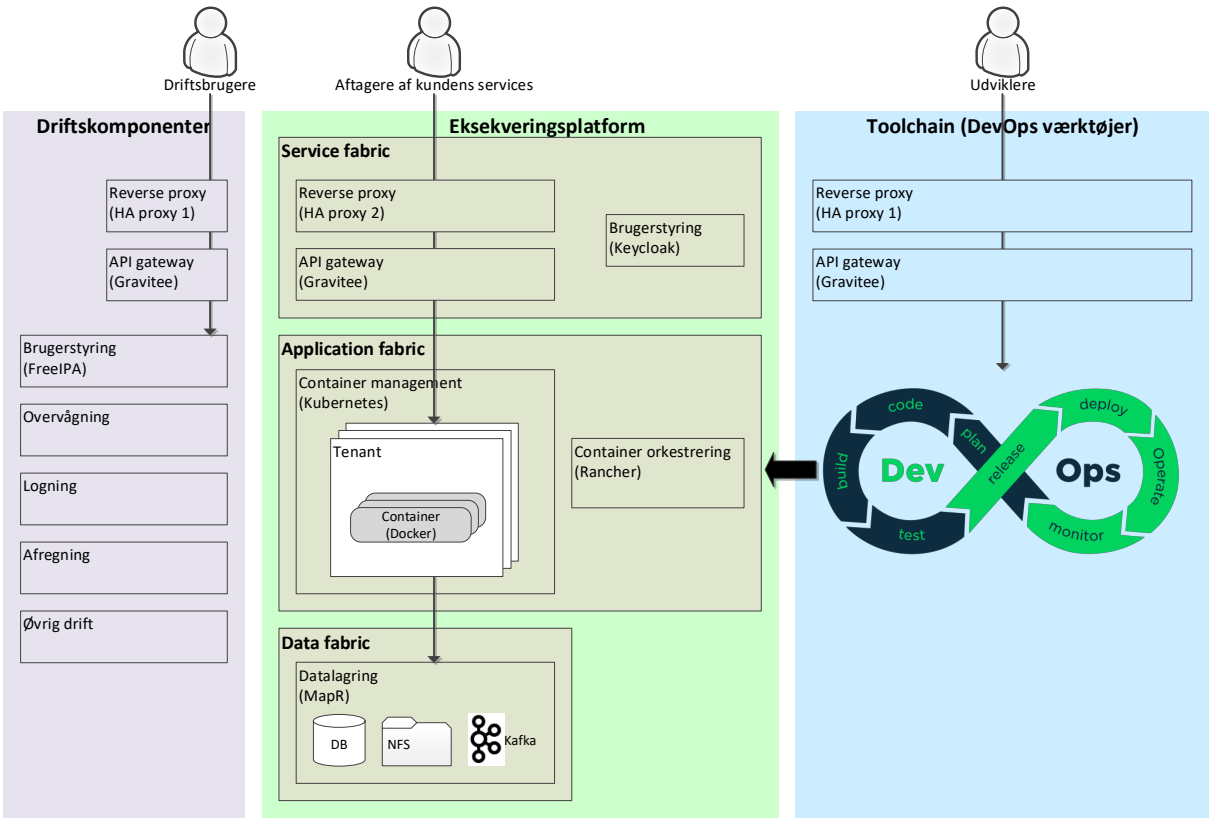
GovCloud er en Platform as a Service (PaaS), som tilbyder cloud-baseret drift vha. container-teknologi. GovCloud tilbyder desuden en række værktøjer (toolchain), som kunderne kan benytte til at udvikle applikationer, eventuelt efter DevOps-metoden.

GovCloud består af mange forskellige komponenter, som primært bygger på open source-produkter. Platformen indeholder dog også nogle proprietære produkter, som er bygget på helt eller delvist open source-software.

GovCloud består overordnet af følgende dele:

- **Eksekveringsplatform** – den del, som står for at eksekvere container-baserede applikationer, gemme resultater og data og udbyde services via internettet.
- **Toolchain (DevOps værktøjer)** – værktøjer, som kunderne kan bruge til bl.a. at udvikle, teste og monitorere applikationer, som skal eksekveres på eksekveringsplatformen.
- **Driftskomponenter** – komponenter som benyttes af SIT's driftspersonale til at administrere og drive GovCloud-platformen.

Komponenterne og deres funktion vises i nedenstående figur.



Figur 1 – GovCloud er sammensat af en række komponenter, hvor en del stilles til rådighed for SIT's kunder, så de selv kan udvikle, deploy'e og drifte deres applikationer.

I de følgende kapitler vil de kundevendte komponenter, som indgår i GovCloud-platformen (dvs. eksekveringsplatform og toolchain) blive beskrevet. Driftskomponenterne, som ikke bruges af kunderne (men kun af SIT), vil ikke blive beskrevet i dette dokument.

3 Eksekveringsplatform

Eksekveringsplatformen giver mulighed for at eksekvere container-baserede applikationer, gemme og hente data, udbyde services fra kundeapplikationer til internettet og for at administrere platformen.

GovClouds eksekveringsplatform (også kaldt Compute fabric) består af:

- Service fabric, der muliggør at kundernes applikationer kan tilgås af eksterne serviceaftagere via internettet.
- Application fabric, som står for deployment og eksekvering af container-baserede applikationer, med automatisk skalering efter behov.
- Data fabric, som gør det muligt for applikationerne i application fabric at gemme og hente permanente data i databaser og andre dataservices.

Kritiske komponenter er dublerede og indeholder en høj grad af redundans, hvilket muliggør en meget høj oppetid.

3.1 Service fabric

3.1.1 HAProxy

HAProxy er en transparent reverse proxy, som giver eksterne serviceaftagere adgang til de kundeapplikationer, som udstilles via GovCloud. HAProxy anvendes således til at etablere adgang til interne services.

HAProxy indeholder desuden en loadbalancer, som kan sprede forespørgsler på tværs af flere containere beliggende på én eller flere hosts.

3.1.1 Gravitee.io

Gravitee.io er en API-gateway, som styrer adgangskontrollen til kundernes udstillede applikations-API'er.

Gravitee.io består af:

- Gravitee.io Gateway - den del af Gravitee.io, der implementerer adgangskontrollen. Gateway'en kontrollerer hvem, hvornår og hvordan brugere får adgang til egne API'er. I modsætning til traditionel HTTP-proxy, er gateway'en i stand til at anvende politikker (dvs. regler) på såvel HTTP-anmodninger som svar i henhold til kundens behov. Det betyder, at det er muligt at berige anmodninger og svarbehandling ved at tilføje transformation, sikkerhed og mange andre støttefunktioner.
- Gravitee.io Manager – Gravitee.io's administrationsmodul til administration af brugere og brugerprofiler. Her kan kunderne logge ind og knytte en applikation til en plan, som igen kan

knyttes til et API. En applikation kan således have flere planer, som hver især knytter sig til ét API.

I Gravitee.io er der også mulighed for at uploade API-dokumentation og knytte autorisationer til API'erne, samt sætte kvoter og lignende på brugen af API'et.

Til adgangskontrol og administration af realms anvender Gravitee.io GovClouds secure token service (Keycloak).

3.1.2 Keycloak

KeyCloak er en Secure Token Service (STS), der kan udstede SAML-, OpenID/OAuth- og JWT-tokens til brug i de udstillede services. Keycloak muliggør single sign-on med identitetsstyring og adgangsstyring, og kan således benyttes til at regulere brugernes adgang til kundernes udstillede applikationer.

Til styring og administration af kundernes applikationer, vil KeyCloak indeholde specifikke realms til hver kunde (tenant), med mulighed for opkobling til eksterne autentificeringsservices, såsom NemLogin.

Login til både GovCloud-services og til de services, der udvikles af kunden til eksterne serviceaftagere, kører så vidt muligt gennem en central login-service (Gravitee.io) som bestyres af Keycloak.

3.2 Application fabric

3.2.1 Kubernetes

Kubernetes står for deployment og eksekvering (drift) af kundernes applikationer i GovCloud. Applikationerne afvikles ikke på én host, men på en sværm af hosts, hvor applikationerne ikke er bundet til bestemte hosts.

Kubernetes anvender internt Docker til afvikling af Docker containere. En Docker container er den mindste enhed til eksekvering af en samlet løsning.

Containeren afvikler én proces, og indeholder hele den stak, der skal til, for at køre processen; det vil sige styresystem, middleware, runtime environment mv. Hvis processen fejler eller på anden vis standser, kan Kubernetes selv starte en ny container op. Tilsvarende, kan den enkelte container også skaleres op eller ned, hvis belastningen hhv. overstiger eller kommer under en vis grænse.

I Kubernetes samles Docker-containere i 'Pods'. En Pod er en samlet logisk enhed, bestående af én eller flere containere, der deploy'es sammen. Containerne i en Pod deler namespace, netværk, og andre parametre. Pods har fastsat en række parametre for genstart og ressourceforbrug, som Kubernetes agerer ud fra. Deployment og den til tider komplekse sammenhæng mellem parametre og Pods indbyrdes, styres af Rancher. Som udgangspunkt, for at undgå unødig kompleksitet, arbejdes der i GovCloud som regel med én container per Pod.

3.2.1 Rancher

Parametrene til afvikling af containere styres gennem Rancher, der overfører disse parametre til Kubernetes, der så efterfølgende agerer efter de opsatte parametre, uafhængigt af Rancher. Kubernetes kan i GovCloud tilgås direkte via et command-line tool, kubectl, men administreres også gennem Ranchers user interface/UI eller command line interface/CLI. Rancher understøtter Role Based Access Control (RBAC) mod Kubernetes.

Rancher kan herudover monitorere/overvåge deployments i Kubernetes.

Rancher anvender Keycloak som adgangskontrol.

3.3 Data fabric

Alle data, der ligger i en container, er flygtige (transiente), og vil forsvinde, når containeren genstartes. Containere indeholder således ikke selv permanente (persistente) data, men gemmer alle sine data uden for containeren i data fabric, som deles mellem alle containere i alle miljøer.
Data fabric i GovCloud udgøres af:

- MapR, der er et kommercielt storage system der understøtter blok, file og objekt lagring.

- Ceph, der er et open source-storage system, der ligeledes understøtter blok, file og objekt lagring.
  Ceph har et Backup-system knyttet, hvor der tages daglig backup af volume, som så opbevares i SITs backup infrastruktur.

3.3.1 MapR

MapR, som er en overbygning på Apache Hadoop. I GovCloud tilbyder MapR følgende datalagringsfunktionalitet:

- NFS - Network File System, et distribueret filsystem, der tillader klienter at tilgå filer og volumes over et netværk.
- OJAI - Open JSON Application Interface, en såkaldt “no-SQL” database til strukturerede data. OJAI er baseret på JSON.
- Kafka - Apache Kafka er en distribueret streaming platform.

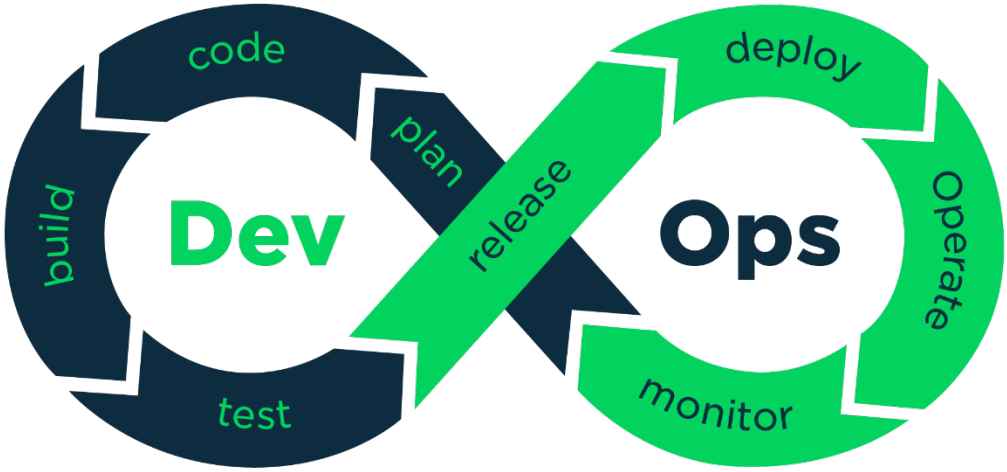
Data ligger ikke på én bestemt harddisk eller én bestemt host, men lagres på en sværm af hosts, hvor data ikke er bundet til bestemte harddiske eller hosts.
MapR kører på 12 fysiske serverer, som synkroniserer data. Denne synkronisering gør at der så godt som ingen risiko er for datatab ved en driftshændelse på storage infrastrukturen.
MapR kan betragtes som et high availability og high output storagesystem.
MapR er dynamisk skalerbar – op og ned. Dvs kundernes applikation dynamisk kan forbruge af den lageringskapacitet der er til rådighed i MapR infrastrukturen. Ligeledes kan der slettes data gemt i MapR. Nærmer forbruget sig den fysisk tilgængelige kapacitet, øges denne proaktivt.
Afregning af MapR lagerforbrug, sker på baggrund af et kvartalsmæssigt snapshot af det aktuelle forbrug, der så anvendes som ”fladt forbrug” for det pågældende kvartal.
3.3.2 Ceph storage

Ceph, er et open source-storage system, der understøtter blok, file og objekt lagring.
Ceph storage allokeres til de enkelte containere/Pods som en fast allokering.
Typisk vil der i en applikation være en ”service pod”, der står for læsning/skrivning til Ceph storage, og der er denne ”service-pod” hvortil der allokeres Ceph storage.
Storage allokering kan enkelt øges via selvbetjening i Kubernetes K8S. Poden skal dog genstartes før den øgede allokering er effektiv.
Opmærksomhed henledes på at applikationen kun kan anvende den allokerede storage, og det er derfor tilrådeligt at der implementeres en overvågning af Ceph storage forbrug i applikationen.
Afregning af Ceph storage sker på baggrund af den allokering der er tildelt
Reduktion af Ceph storage er ikke muligt som selvbetjening i Kubernetes. Reduktion af Ceph storage kan kun foretages som en bestillingsopgave, og denne medfører nedetid for applikationen.

Ceph har et Backup-system knyttet, hvor der tages daglig backup af volume, som så opbevares i SITs backup infrastruktur.

4 Toolchain (DevOps værktøjer)

GovCloud udstiller en toolchain (en samling af værktøjer) som understøtter DevOps. DevOps er en sammentrækning af Development og Operations (udvikling og drift), og er en metode, som understøtter de processer som indgår i udvikling og drift, fra planlægning, over udvikling og test, til produktion og overvågning.



Figur 2 - GovClouds toolchain understøtter alle led i applikationsudviklings- og applikationsdriftsprocesserne.

De kunder som bruger GovCloud, kan vælge helt eller delvist at benytte GovClouds toolchain til at udvikle, teste og monitorere applikationer, som skal eksekveres på GovCloud. De kan også benytte deres egen toolchain.

Værktøjerne i GovClouds DevOps toolchain understøtter følgende udviklings- og driftsprocesser:

- Planlægning
- Kodning, build og test
- Release og deployment
- Drift og overvågning

4.1 Planlægning

I planlægningsfasen defineres og dokumenteres de opgaver, behov og krav som løsningen skal løse, så der løbende kan følges op på fremdriften. Ud fra et vidensdelingsperspektiv kan det desuden være en fordel at dokumentere beslutninger og designløsninger for at gøre dem lettilgængelige for alle, som på et tidspunkt skal arbejde med løsningen.

4.1.1 Jira

Jira er et projektstyringsværktøj, som understøtter mange forskellige projektstyringsmetoder, såsom Scrum, Kanban og vandfaldsmodellen og som kan bruges til planlægning, problemsporing og ressourcestyring. Jira kan også benyttes til almindelig driftsmæssig behandling af f.eks. opgaver, problemer og fejl.

Jira anvender Keycloak som adgangskontrol.

4.1.2 Confluence

Confluence er en Wiki, dvs. et værktøj til dokumentation og vidensdeling. Det er nemt at søge relevant information frem, og der kan laves hyperlinks til andre dokumenter. Confluence har desuden et tegnemodul, som kan bruges til at udarbejde forskellige typer af tegninger.

Dokumentationen i Confluence er som udgangspunkt offentlig for alle, som har adgang til et specifikt arbejdsområde (rum/space). Alle kan som udgangspunkt rette i alt. Derved bliver det nemmere at løbende vedligeholde dokumentation og holde den relevant.

Der er versionskontrol på alle dokumenter, så det er muligt at se, hvilke rettelser, der er foretaget. Det er også muligt at gå tilbage til en tidligere dokumentversion eller lave midlertidige rettelser uden at udgive disse, for så senere at publicere.

Confluence har en tæt kobling til Jira og det er f.eks. muligt at oprette projekter i Confluence og herfra linke direkte til sager i Jira.

Confluence anvender Keycloak som adgangskontrol.

4.2 Kodning, build og test

GovClouds toolchain stiller værktøjer til rådighed for bl.a. kildekodeintegration, test, versionsstyring, build, dokumentation, og issue tracking.

Til understøttelse af kildekodeintegration udbyder GovCloud værktøjer, der understøtter merge requests og systematisk review af kode, hvor deltagere i et udviklings-team kan vurdere ændringer i kildekoden, give feedback til udvikleren og enten godkende eller afvise ændringerne.

Til understøttelse af automatiseret test, kan en build server være anvendelig. Et komplet miljø med testdata kan deploy'es og startes automatisk, og fejlende tests meldes tilbage til de udviklere, der har foretaget ændringerne i den testede version af softwaren.

4.2.1 GitLab

GitLab tilbydes som software repository med indbyggede funktioner for versionsstyring, dokumentation og issue tracking samt indbygget CI/CD[1] pipeline.

I GovClouds toolchain tjener GitLab flere formål. GitLab bestyrer først og fremmest det centrale Git repository og stiller funktionalitet til rådighed til at administrere dette repository. Disse funktioner dækker bl.a. over:

- Adgangskontrol – som sikrer, at identiteten af kilden til ændringerne er kendt.
- Brugeradministration – som sikrer, at kun de af kunden godkendte ressourcer (brugere, systemer etc.) kan foretage ændringer.
- Repository browser – som tillader visning af ændringer til kildekode-review uden anden installation af software, end en browser.
- Merge request – som er en metode til forespørgsel om integration af en software-ændring i kildekoden, og som kan danne grundlag for et kode-review af ændringen.

GitLab indeholder desuden GitLab Runner, som indeholder en del funktionalitet, der til dels er dækket af andre værktøjer, men som her tilbydes i en letvægtsudgave. Disse inkluderer bl.a.:

- Problemhåndtering/Issue tracking - dette overlapper noget med Jira, men GitLab Runner er simplere og nemmere at tilgå, dog på bekostning af reduceret fleksibilitet.
- Build server - denne funktionalitet overlapper med Jenkins, men GitLab Runner er enklere og mere direkte integreret med GitLab.

Adgang til det centrale Git repository i GovCloud (GitLab) foregår ved hjælp af SIT's B- og X-konti og GovClouds STS (Keycloak). Herefter skal der tilføjes en SSH-nøgle alternativt en adgangskode til basic authentication (https).

4.2.2 Git

Git anvendes til versionsstyring af det kildemateriale eller de artefakter, der udarbejdes manuelt af udviklingsteamet, såsom f.eks. kildekode, grafik, websider, skabeloner, konfigurationsfiler, versionsafhængigheder, build jobs og scripts.

Git kan vise ændringer i et udviklingsforløb, vise hvem der har lavet hvad og der kan søges på, hvilke ændringer der er foretaget tættest på introduktion af en fejl. Git bliver på den måde ikke blot et redskab til versionsstyring, men et redskab til at kommunikere mellem udviklere om ændringer.

[1] Continuous Integration (CI) og Continuous Delivery (CD), er en praksis hvor udvikleren løbende (flere gange om dagen) integrerer kode til et fælles arkiv (repository) som straks, efter automatiseret test og build, bliver deploy'et i produktion.

Git kan håndtere såvel tekstbaseret kildemateriale, som binære filer (dvs. billeder, ikoner mv.). Selvom Git kan håndtere binære filer, så bør Git dog udelukkende anvendes til at lagre materiale, der er udviklet i projektet og ikke kan genskabes ad anden vej.

Git bør ikke anvendes til materiale, der er hentet fra nettet, binære moduler, færdig-build'ede komponenter, container images osv. Til disse artefakter er andre værktøjer som f.eks. jFrog Artifactory og Harbor bedre egnede.

Git er et produkt, der installeres på de enkelte udviklings-pc'er, som en del af den lokale udviklingsplatform. Det er således ikke et produkt GovCloud leverer.

Adgang til det centrale Git repository foregår gennem GitLab.

4.2.3 Jenkins

Jenkins er en dedikeret build server, der anvendes sammen med dependency management[2], build scripts, compilere og deployment-systemer til at oversætte kildekode og andre artefakter, til færdige applikationer.

Med Jenkins kan der opstilles jobs, der både build'er, gemmer build'ede komponenter i et artefaktrepository, samler og deploy'er komplette midlertidige miljøer, udfører automatiske tests og rapporterer om fejlende build eller test til de ansvarlige (dvs. de udviklere der har lavet ændringer i koden).

Jenkins (eller GitLab Runner) bliver ofte et centralt punkt for udvikling efter DevOps-metoden, da det vil være herfra al build, test, og deployment finder sted.

Jenkins anvender Keycloak som adgangskontrol.

4.2.4 jFrog Artifactory

jFrog Artifactory er et build repository, der anvendes til at gemme færdig-build'ede komponenter, enten egenudviklede komponenter, eller komponenter der er hentet fra andre repositories.

Når artefakter build'es fra kildekode, er der ofte afhængigheder til andre artefakter. Artefakter, der enten hentes fra internettet, eller som blot kan build'es igen fra kildekode. Disse artefakter eller halvfabrikata gemmes under versionskontrol i jFrog Artifactory, for hurtigere at kunne anvende dem igen og igen, uden at skulle downloade eller starte build'en forfra hver gang.

jFrog Artifactory anvender sin egen adgangskontrol, og brugerne skal oprettes gennem en administratorkonto.

[2] Anvendelsen af et artefakt repository (se jFrog Artifactory), navngivning af komponenter, referencer til navngivne versioner m.v., hører under disciplinen "Dependency Management", og styres ofte med build værktøjer som Gradle, Maven eller Ivy, som Jenkins understøtter.

4.3 Release og deployment

I en release (frigivelse), udvælges en bestemt version af applikationens underliggende software, til at definere en ny applikationsversion, hvilken herefter kan deploy'es (idriftsættes/installeres) i et miljø (fx et udviklings-, test- eller produktionsmiljø). Release og deployment-processen kan enten være manuel eller fuldt automatiseret.

En (produktions)release er en færdig version af softwaren, som kan sættes i produktion og benyttes af slutbrugerne.

4.3.1 Harbor

Harbor er et container repository til Docker images, hvor container images lagres for senere deployment på en eller flere ekskveringsplatforme. Images i Harbor kan danne grundlag for nye images, eller deploy'es direkte i Kubernetes enten via Rancher eller direkte via kubectl (Kubernetes kommandoprompt).

Når en løsning i GovCloud sættes sammen til en container-baseret service, består denne løsning af én eller flere containere, der hver for sig indeholder en komplet stak af software til én komponent, en pod. En sådan samlet stak, kaldet et image, gemmes i et særligt repository, så det let kan deploy'es i et miljø senere, uden at build'e komponenterne igen.

Harbor anvender Keycloak som adgangskontrol.

4.3.2 Rancher

Se afsnit 3.2.1 Rancher.

4.4 Drift og overvågning

Efter at en version er blevet deploy'et, afvikles applikationen i Kubernetes med de parametre, der er opsat via kubectl eller Rancher. I drift kan kunderne selv opsætte logning og overvågning af applikationen. Toolchain'en tilbyder desuden værktøj til koordinering og vidensdeling samt performance- og fejlanalyse.

4.4.1 Mattermost

Mattermost er et chat-program, som kan benyttes til koordinering og vidensdeling. Konversationer i Mattermost kan gemmes og bruges som vidensdatabase for f.eks. incident-løsningsforløb. Det er muligt at oprette private konversationer, som kun kan ses og benyttes af de brugere, der gives adgang.

Mattermost har egen adgangskontrol med oprettelse af bruger-id (e-mailadresse) og password.

4.4.1 Zabbix

Zabbix er et overvågningsværktøj til flere it-komponenter, herunder netværk, servere, virtuelle maskiner og cloud-services. Zabbix leverer overvågningsmetrikker til blandt andet netværksudnyttelse, CPU-belastning og diskpladsforbrug.

4.4.2 Grafana

Grafana visualiserer data i form af dashboards med tabeller og grafer. Grafana bruges i GovCloud til at udstille de overvågningsdata, som indsamles af Zabbix.

4.4.1 ElasticSearch

ElasticSearch er et log management-værktøj, der indsamler en del af de lokale logs, som GovCloud-platformen genererer. ElasticSearch indeholder også en søge- og analysemotor til logdata. Analyser af logdata kan dog med fordel foretages i Kibana.

4.4.2 Kibana

Kibana er et log-management-værktøj og en grafisk front-end til ElasticSearch. Med Kibana kan logdata i ElasticSearch visualiseres med diagrammer og grafer.

4.4.3 Jaeger

Jaeger (Tracing) kan benyttes til overvågning og fejlanalyse af servicebaserede, distribuerede systemer og understøtter f.eks.:

- Distribueret transaktionsovervågning
- Optimering af ydelse og reaktionstid
- Root cause analyse
- Serviceafhængighedsanalyse.