

SORBONNE UNIVERSITÉ
FACULTÉ DES SCIENCES & INGÉNIERIE
INGÉNIERIE DE L'INFORMATIQUE INDUSTRIELLE ET DE L'IMAGE

POC d'une solution de partage de fichiers volumineux à la ville de Paris

Mémoire de deuxième année master en apprentissage I4

Effectué au sein du Bureau de l'Intégration applicative et du DevOps
Mairie de Paris

Présenté et soutenu par
Yahia IHDENE

Sous la direction de
M. Jean-Marc LAFON

Date de soutenance
18 septembre 2020

Remerciement

Je remercie tout d'abord la Ville de Paris, ainsi que Mme Lydia MELYON, chef du Bureau de l'Intégration Applicative et du DevOps (BIAD), de m'avoir accueilli cette année pour mon alternance. Cela m'aura permis d'acquérir de l'expérience en développement et intégration et également de découvrir, plus en profondeur, le travail en milieu professionnel.

Je tiens particulièrement à remercier mon Maître d'apprentissage M. Jean-Marc LAFON, Architecte au sein du Bureau de l'Intégration Applicative et du DevOps, pour son soutien, son aide et la confiance qu'il m'a accordée tout au long de ma formation.

Je souhaite également remercier Mme Carole BELLAICHE, M. Pierre Gabrielle et M. El-Mostafa ZOUGAR pour leurs aides et soutiens lors de la réalisation des missions qui m'ont été confiées.

Une pensée particulière est enfin adressée à l'ensemble des membres de l'équipe du BIAD qui n'a pas hésité à me proposer leur aide pour la rédaction de ce travail.

Glossaire

BIAD	Bureau de l'Intégration Applicative et du DevOps.
DevOps	Pratique technique visant à l'unification du développement logiciel (dev) et de l'administration des infrastructures informatiques (ops).
STIPS	Service Technique des outils numériques, des Infrastructures, de la Production et du Support.
DSIN	Direction des Systèmes d'Information et du Numérique.
MDP	Mairie De Paris.
SFTP	(Secure File Transfert Protocol) permet de transférer des fichiers par une connexion sécurisée utilisant le protocole communication sécurisé SSH.
POC	Proof Of Concept (Preuve de concept).
NFS	Network File System.
LDAP	Protocole qui permet d'interroger les services d'annuaire des services.
WSSO	est une méthode permettant à un utilisateur d'accéder à plusieurs applications informatiques en ne procédant qu'à une seule authentification.
SAML	Security Assertion Markup Language (est une norme ouverte pour l'échange de données d'authentification et d'autorisation entre les parties).
RBAC	Role-Based Access Control (un modèle de contrôle d'accès dans lequel chaque accès est basée sur le rôle auquel l'utilisateur est associé).
Cluster	Groupe de serveurs qui forme une unité logique unique agissant comme un seul et même système redondant.
PostgreSQL	Bases de données relationnelle.
MariaDB	Bases de données relationnelle.
Redis	Serveur de structures de données.
Firewall	Un logiciel et/ou un matériel permettant de faire respecter la politique de sécurité du réseau.
NetScaler	Proxy inverse répartiteur de charge.
Scalable	capacité à s'adapter à une montée en charge

Résumé

Dans le cadre du développement et de l'amélioration des outils de communication de la DSIN, la direction décide de mettre en place un POC NextCloud (preuve de concept d'un cloud privé basé sur la solution NextCloud). Ce POC permettra d'étudier la faisabilité du projet d'un point de vue technique et les améliorations qu'il pourra apporter aux outils de communication de la ville de Paris.

Ce mémoire présente une solution scalable d'échange de fichiers volumineux destinée à environ 15 000 usagers et les différentes architectures de l'application selon l'environnement de développement, la méthodologie suivie pour la mise en place de toutes les briques composant la solution ainsi que les tests appliqués pour déterminer la faisabilité de notre solution puis nous terminerons avec une synthèse des résultats obtenus à l'issue de ces tests et la suite qui s'offre au projet.

Mots clés : Intégration et déploiement d'une application, mise en place d'un POC, étude de faisabilité, amélioration des services.

Abstract

As part of the development and improvement of DSIN's communication tools, the management decided to implement a NextCloud POC (proof of concept of a private cloud based on the NextCloud solution). The main purpose of developing a POC is to demonstrate the functionality of the project from a technical standpoint and the improvements it will be able to make to communication tools.

This training report presents a scalable large file exchange solution intended for approximately 15,000 users and its different architectures following the development environment, the methodology followed for the implementation of all the bricks composing the solution and all the tests applied to determine the feasibility of our solution then we will finish with a synthesis of the results obtained at the end of these tests and the next step for the project.

Keywords: Integration and deployment of an application, implementation of a POC, demonstrate the functionality, service improvement.

Table des matières

I-	<i>Introduction générale</i>	1
II-	<i>Synthèse du travail réalisé en 1^{ère} année d'apprentissage</i>	2
III-	<i>Présentation de l'entreprise</i>	4
1-	Présentation de l'administration	4
2-	Présentation de la DISN	4
a-	Service de la Transformation et de l'Intégration Numériques STIN	4
b-	Sous-Direction des Ressources SDR	4
c-	Service Technique des outils numériques, des Infrastructures, de la Production et du Support STIPS....	4
3-	Organigramme de la DSIN	5
a-	Organigramme du BIAD	6
IV-	<i>Présentation du projet</i>	6
V-	<i>Méthodologie du projet</i>	8
VI-	<i>Intégration du POC NextCloud</i>	9
1-	Briques fondamentales	10
a-	Installation d'Apache et PHP ^[3]	10
b-	Installation de NextCloud ^[3]	10
c-	Installation de PostgreSQL	11
2-	Intégration du POC en DEV	12
a-	Déploiement de la base de données	12
b-	Configuration de l'application NextCloud	13
3-	Intégration du POC en REC	14
a-	Mise en place de la réplication PostgreSQL ^[1]	14
b-	Configuration du mode réplication	14
c-	Installation de Redis ^[7]	15
d-	Montage NFS ^[5]	16
4-	Intégration du POC en Préproduction	18
a-	Sécurisation de l'application ^[6]	18
b-	Customisation du POC ^[4]	20
VII-	<i>Installation de Pgwatich</i>	23
1-	Installation du serveur base de données	23
2-	Installation du serveur applicatif ^[8]	24
3-	Configuration du serveur de base de données	24
4-	Configuration du serveur applicatif	26
VIII-	<i>Tests et résultats</i>	27
IX-	<i>Conclusion</i>	30
X-	<i>Bibliographie</i>	31
XI-	<i>Table d'annexes</i>	32

I- Introduction générale

La Révolution numérique a peut-être relié les continents en les connectant grâce à des messageries instantanées et des courriers électroniques et a permis aux utilisateurs de sauvegarder et d'échanger des informations via leurs ordinateur et smartphone, mais quelques fois, ils se retrouvent face aux limites de ces systèmes d'échanges telle l'impossibilité de sauvegarder et/ou d'envoyer des fichiers volumineux.

Le développement remarquable du Cloud, ces dernières années, suscite de plus en plus l'intérêt des différents utilisateurs d'informatique qui cherchent à profiter au mieux des services et des offres disponibles en ligne à travers le web. Ce qui incite également la Direction de System d'Information et du Numérique (DSIN) à se demander « **comment mettre en place un service cloud permettant aux agents de la Ville de Paris de sauvegarder et d'envoyer des fichiers volumineux en toute sécurité** ».

Pour répondre à cette problématique, la DSIN décide alors, de mettre en place un POC basé sur la solution cloud de NextCloud pour démontrer la faisabilité du système notamment sur la qualité du service, la disponibilité de l'application et ses performances, avant d'aller demander une prestation à une entreprise tierce pour la mise en place de l'application finale.

Cette solution permettra à la DSIN de contrôler de manière directe les données de la Ville de Paris en ayant les données hébergées dans ses data center et d'avoir une grande sécurité vis-à-vis des communications et des échanges de données sensibles entre le personnel. Elle permettra également aux agents de la ville de synchroniser leurs données, fichiers, contacts, agendas entre différents appareils et de pouvoir échanger entre eux des fichiers volumineux.

Pour l'ensemble de la réalisation et la mise en place du POC, nous avons procédé par trois phases principales. La première est d'appliquer la démarche de la mise en place de l'application sur un environnement de développement puis sur un environnement de recette afin de faire une analyse fonctionnelle et une étude des caractéristiques du produit, la deuxième est d'appliquer la démarche de la mise en place de l'application sur un environnement de préproduction pour une analyse de performances en mesurant le temps de réponse de l'application en tenant compte du volume de trafic attendu et ceci en s'appuyant sur l'application tierce « Pgwach » qui est une application de supervision dédiée aux mesures de performances, pour finir la dernière étape est la phase pilote prévue en mi-septembre 2020 pour voir concrètement les résultats obtenus, si le POC répond aux exigences de la Ville de Paris, la DSIN passera une demande de prestation à l'entreprise spécialisée dans l'intégration de l'application en production et sa publication en externe.

Ce mémoire a pour objet de présenter l'intérêt et le rôle qu'a le projet NextCloud sur l'amélioration que la ville de Paris a comme objectif, la méthodologie suivie pour la mise en place et la supervision de toutes les briques de l'application, nous allons également expliquer comment NextCloud permet de faciliter les échanges entre les agents de la MDP et de synchroniser les données, fichiers, contacts, agendas entre différents appareils

En premier lieu, nous présenterons l'application NextCloud sur laquelle s'appuie notre projet, nous expliquerons également les différentes briques qu'elles le composent, la procédure d'installation et d'intégration de ces dernières.

En second lieu, nous exposerons la deuxième application essentielle à la supervision « Pgwatch » qui vient accompagner notre projet pour superviser le bon fonctionnement des bases de données de NextCloud.

En dernier, nous proposerons une synthèse et une conclusion du travail fait à l'égard de notre problématique, les résultats obtenus du POC mis en place (Go ou No GO) en fonction des besoins de La ville de Paris et la suite potentielle qui s'offre à notre projet NextCloud en fonction des résultats obtenus.

II- Synthèse du travail réalisé en 1^{ère} année d'apprentissage

Durant ma première année d'apprentissage au sein du Bureau de l'Intégration Applicative et du DevOps, j'ai eu pour mission la mise en place l'architecture technique et l'intégration de l'application Botpress de l'environnement de développement jusqu'en production puis de déployer en collaboration avec le Bureau de l'ingénierie logicielle et du développement un chatbot pour le site Facil'famille de la Ville de Paris pour remédier à la problématique posée qui est « les possibilités offertes par les sites web de la ville sont nombreuses et parfois difficiles à appréhender. L'utilisateur a besoin d'être accompagné ».

Au début du mois de juin 2019 la DSIN affecte la tâche de l'intégration de plate-forme Botpress au BIAD, dès mi-juin 2019 en recevant les machines nécessaires nous avons commencé à installer et à configurer l'application en environnement de recette. Suite à la validation de l'application nous avons mis en place une base de données qui a pour but de contenir toutes les informations et les configurations des chatbots.

En amont de la mise en place de la recette, nous avons commencé à préparer l'environnement de production, notamment en faisant les demandes des machines virtuelles, en installant l'application Botpress, en faisant les demandes nécessaires comme les demandes d'url d'accès interne/externe et en mettant en place l'application Lutèce avec l'aide du Bureau de l'ingénierie logicielle et du développement du qui jouera le rôle d'intermédiaire entre l'utilisateur et le back-end (Botpress).

Nous avons eu le retour des demandes d'accès interne/externe en fin du mois de juillet 2019 ce qui nous a permis de commencer les tests du bot et les corrections des bugs. Nous avons également en mois d'août 2020 commencés à intégrer la première version du bot Facil'Famille en recette afin de faire des tests sur le temps de réponse, l'amélioration du débogueur et d'enrichir le chatbot avec des questions grâce à ses capacités d'apprentissage.

En fin du mois de septembre nous avons commencé à finaliser la mise en place de l'application Botpress en haute disponibilité sur l'environnement de production. Nous avons importé le chatbot que nous avons déjà préparé en amont sur l'environnement de recette et nous l'avons intégré dans l'environnement de production. Suite à cela nous avons mené une phase de test pour garantir le bon fonctionnement du chatbot Facil'Famille en posant des questions que nous connaissons déjà les réponses afin de comparer les réponses avec les résultats attendus. Nous avons également mis en place un proxy inverse avec fonctionnalité de load-balancer pour la gestion du flux d'utilisateurs.

La publication pour le grand public du bot Facil'Famille a été faite en octobre 2019. Cela a été fait après avoir eu la licence du produit Botpress, ouverture des flux en externe par le service qui a pour mission la gestion d'identité et sécurité de la ville et une montée de version sur le produit Botpress. Ma mission à partir de ce moment était le transfert de compétences vers la Section Intégration des Services Numériques qui s'occupe de maintenir l'application en marche en cas d'indisponibilités, faire des montées de version si nécessaire et de corriger les éventuels bugs sur l'application.

Le transfert de compétences a été fait via des réunions où mon rôle était d'expliquer le mode de fonctionnement de l'application, les points sensibles et la méthodologie à suivre en cas où les membres de la section désirent effectuer des manœuvres sur l'application. Les réunions étaient accompagnées par des documents qui expliquent en détail l'application et les démarches à suivre. Parmi les documents nous avons le Document d'architecture Technique (DAT) qui explique l'architecture de toute l'application, le Dossier d'Installation (DI) qui explique comment installer ou faire une montée de version sur l'application, le Dossier d'Exécution (DEX) qui explique comment exécuter l'application et les programmes qu'ils l'accompagnent et également des consignes d'arrêt/relance en cas où ils devront arrêter les services de l'application, de les relancer et de vérifier leur bon fonctionnement.

L'application finale a été publiée en externe sous le l'URL suivant : <https://bot.paris.fr/>

III- Présentation de l'entreprise

1- Présentation de l'administration

La Ville de Paris est une des plus grandes administrations publiques de France. En effet, la ville est divisée en 20 arrondissements, chacun dirigé (administrativement) par un maire.

De plus, chaque arrondissement se voit élire des conseillers de Paris (au minimum 3 par arrondissement), siégeant au conseil de Paris. Cette assemblée, possédant les attributions d'un conseil municipal et d'un conseil départemental, est présidée par la Maire de Paris.

La Ville de paris possède donc une structure assez large et complexe. En quelques chiffres, elle regroupe plus de 50 000 agents, 4 grands pôles, 23 directions et environ 3 000 sites et parmi les 23 directions, nous avons la Direction des Systèmes de l'Information et du Numérique « **DSIN** ».

2- Présentation de la DISN

La DSIN est responsable de l'ensemble des composants matériels et logiciels du système d'informations (serveurs, équipements de réseau, systèmes de stockage...) et des 850 applications de la ville de Paris.

Elle est chargée de maintenir la sécurité, la fiabilité des systèmes, de développer et de mettre en œuvre au bénéfice de l'ensemble des directions de la Ville et du Département, les systèmes de traitement et de transmission de l'information. Elle accompagne également l'ensemble des directions métiers dans l'exercice de leurs missions.

La DSIN se devise en trois services qui sont :

a- Service de la Transformation et de l'Intégration Numériques STIN

Chargé de l'administration des données, la conception et la conduite de projet mais aussi de la maintenance et du développement des applications.

b- Sous-Direction des Ressources SDR

Chargé de la gestion du budget et des ressources humaines, de la communication interne et du recrutement et elle est en relation avec tous les agents de la Direction des Système Informatique et du Numérique.

c- Service Technique des outils numériques, des Infrastructures, de la Production et du Support STIPS

En charge de fournir aux agents de la Ville leur environnement numérique de travail et de les assister dans leur usage. Il est aussi en charge de concevoir et de maintenir le socle technique sur lequel reposent les applications métiers ainsi que des services numériques à destination des Parisiens. Il est enfin en charge de la conception et du maintien des infrastructures portant l'ensemble de l'offre de service informatique et numérique interne et externe de la Ville de Paris (infrastructures de sécurité informatique, réseau et télécommunication).

Parmi les bureaux qui composent le STIPS on trouve le **BIAD**, le Bureau de l'Intégration Applicative et du Devops, qui a pour mission d'accompagner la conception technique des outils numériques et des applications et la prise en charge des activités d'intégration applicative suivant une démarche d'industrialisation pour l'ensemble des bureaux DSIN ou partenaires dont les applications sont hébergées par la DSIN.

Pour finir le BIAD se devise trois en section, la Section FAST DATA & PAAS (SFD&P), Section Intégration des Services Numériques (SISN) et la Section Étude Ingénierie Intégration Applicative (EI²A) dans laquelle moi et mon maître d'apprentissage faisons partie.

Notre rôle en tant qu'Architecte est de mettre en place l'architecture technique de nouvelles applications, d'assurer leur bonne intégration avec toutes les briques du système d'information de la ville de Paris.

3- Organigramme de la DSIN

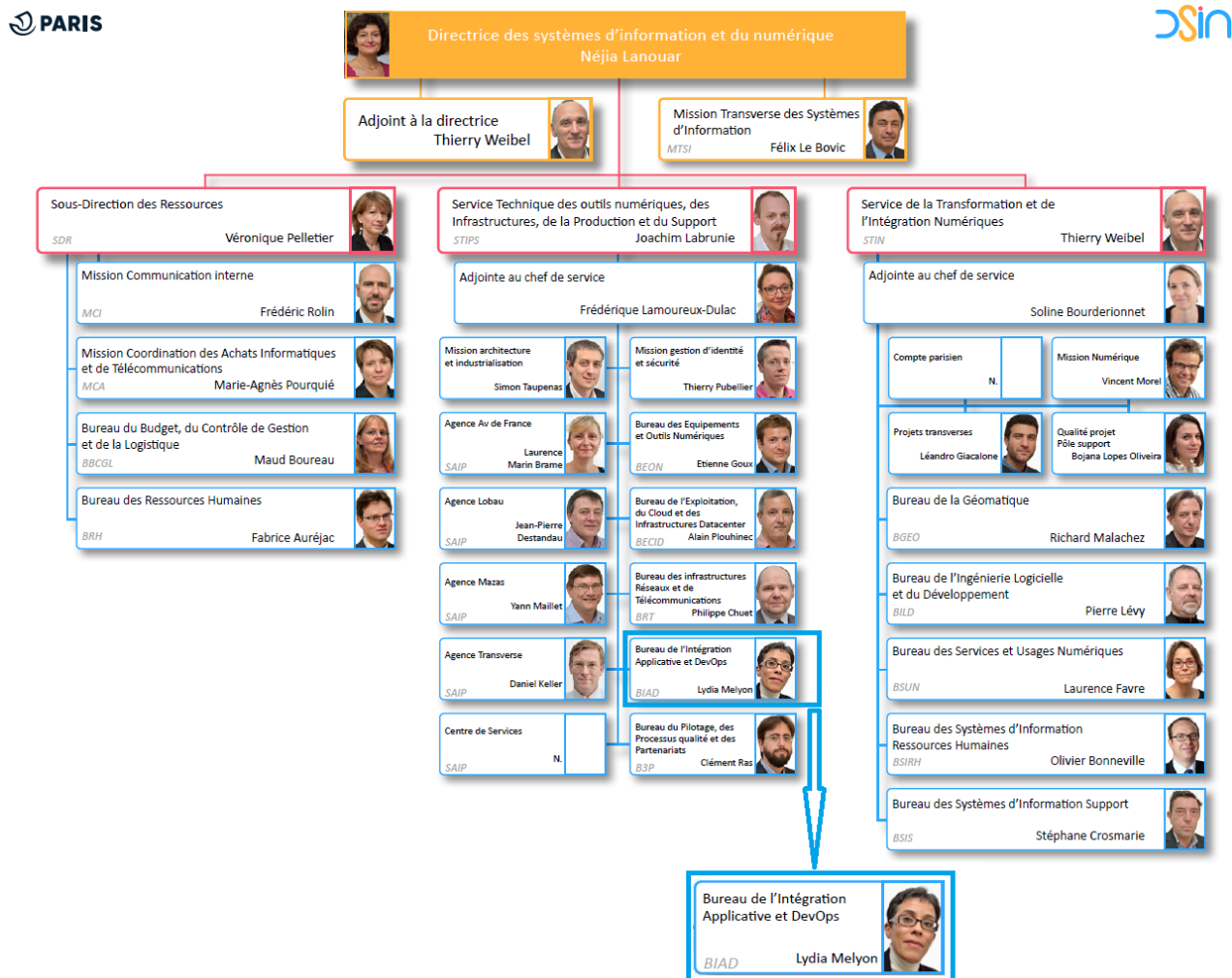


Figure 1 : Organigramme de la DSIN

a- Organigramme du BIAD

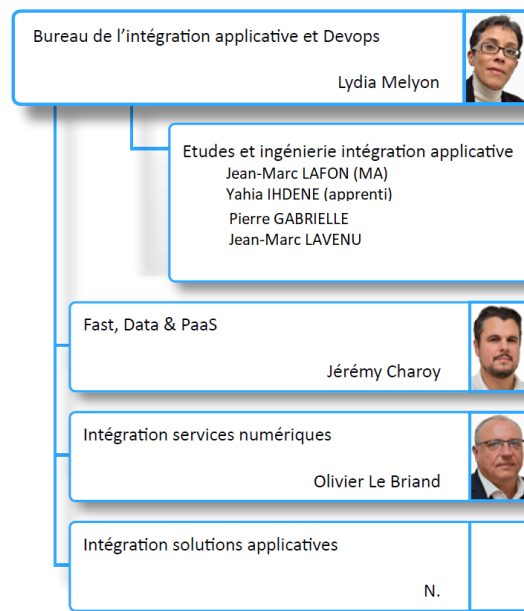


Figure 2 : Organigramme du BIAD

IV- Présentation du projet

Les agents de la ville de Paris communiquent quotidiennement et échangent des documents et des informations par mails, parfois les boîtes mail n'arrivent pas à assurer la transmission de certaines informations essentielles, notamment quand il s'agit de fichier volumineux.

Fournir aux agents de la Ville de Paris un système qui compense les limites des systèmes de communication classiques est fondamental pour garantir de meilleures conditions de communication entre les agents.

Après une étude des différents produits présents sur le marché qui peuvent améliorer les services et éventuellement résoudre la problématique, la DSIN a fini par choisir le produit NextCloud qui est une solution open-source privilégiée dans la fonction publique.

NextCloud est une suite de logiciels client-serveur permettant de créer et d'utiliser des services d'hébergement de fichiers. NextCloud est gratuit et open source, l'intérêt de NextCloud est le fait de pouvoir l'utilisée sur des serveurs privés pour l'hébergement des données, permet également l'administration des utilisateurs et des groupes (fédération d'identité SAML couplé à une RBAC interne à l'application). Le contenu peut être partagé en définissant des droits de lecture/écriture granulaire entre les utilisateurs et les groupes. Les utilisateurs de Nextcloud peuvent également créer des URL publiques lors du partage de fichiers pour les personnes possédant pas un compte NextCloud.

La solution étant modulaire, elle peut être étendue avec des plugins pour implémenter des fonctionnalités supplémentaires.

Ma mission en tant que Architect apprenti au BIAD avec l'aide de mon maître d'apprentissage M. Jean-Marc LAFON est de mettre en place le POC de l'application NextCloud pour pouvoir étudier la faisabilité de l'application et de répondre aux questions sous-jacentes qui sont essentiels pour trancher sur la publication définitive de l'application telle « est-ce que la solution retenue est la bonne ? » , « Est-on capable de la mettre en œuvre ? » , « Quels sont les risque et le niveau de sécurité de l'application ? ».

La mise en place du POC de l'application NextCloud ainsi que des applications sur lesquelles il se repose sont faites en plusieurs étapes.

La première étape est faite en deux phases, la première est l'installation de l'application dans un environnement de développement dans laquelle se trouvent juste les parties essentielles du projet notamment un serveur web (apache + PHP 7.2) un serveur de base de données PostgreSQL et un serveur Redis. La deuxième phase est l'installation de l'application dans un environnement de recette dans laquelle nous avons repris ce que nous avons fait en développement et nous avons rajouté des répliquions sur les serveurs pour garantir une haute disponibilité ainsi qu'un serveur de fichier dans lequel nous avons stocké tous les informations des utilisateurs et les données qu'ils partagent entre eux. Nous avons mis deux serveurs web avec un répartiteur de charge (Load-balancer), un cluster de base de données PostgreSQL avec deux serveurs Actif/Passif et un cluster Redis avec 2 serveurs Actif/Passif.

Le but de cette première installation est purement fonctionnel, c'est-à-dire que l'objectif visé par la démarche est de vérifier que NexCloud peut satisfaire les besoins des agents de la ville.

La deuxième étape est l'installation du POC dans un environnement de préproduction dans lequel se trouvent toutes les briques du projet avec toutes les répliquions nécessaires pour un bon fonctionnement. Le projet se compose de quatre serveurs web (Apache + PHP 7.2) avec un répartiteur de charge (Load-Balancer), d'un cluster de base de données PostgreSQL avec deux serveurs Actif/Passif, d'un cluster Redis avec deux serveurs Actif/Passif et d'un serveur de fichier en NFS d'une taille de 1 To pour stocker les fichiers volumineux qui transite grâce à NextCloud. Nous avons également mis en place une application de supervision Pgwach qui nous permet d'avoir des données concrètes sur l'état de nos serveurs de bases de données pour étudier leurs performances.

L'application Pgwach se compose également d'un serveur web dans lequel nous avons l'application Grafana qui va nous permettre de visualiser les données sous forme de graphe, une base de données PostgreSQL de configuration Pgwach et InfluxDB, une base de données des métriques et des données de configuration Grafana.

Le but de cette deuxième installation (NextCloud en préproduction et Pgwatch) est dans le cadre de l'étude des performances des serveurs en fonction des requêtes que les utilisateurs envoient et cela en se basant sur les données que Pgwatch nous fournis.

À l'issue des résultats obtenu à la phase pilote en préproduction, la DSIN décidera de faire ou pas en septembre une intégration de l'application avec la prestation de d'une entreprise externe dans le cadre d'une architecture plus scalable en fonction de la demande pour la production.

V- Méthodologie du projet

L'équipe autour du projet est composée de développeurs du Bureau des équipements et outils numériques (BEON) et d'architectes du BIAD dont moi et mon maître d'apprentissage faisant partie.

Dans le cadre de l'installation et l'intégration du POC NextCloud dans les SI de la Ville, nous avons instauré une démarche de travail dite " méthode Agile" qui nous a permis l'organisation de projet, la gestion des différentes tâches et nous a garanti un bon suivi de l'évolution du projet.

La méthode utilisée dans notre pour la gestion de projets était la méthode Agile SCRUM qui fait partie de la famille agile. L'approche consiste à se donner des objectifs à courts terme, une fois l'objectif terminé nous faisons le point avec l'équipe du BEON et suivant les résultats obtenus précédemment nous fixons les nouveaux et ainsi de suite jusqu'à atteindre le résultat final. Le commanditaire est impliqué dans le projet du début à la fin ce qui lui donne de la visibilité.

Cela permet entre autres d'avoir une grande capacité d'adaptation au changement grâce à des itérations courtes, d'éviter « l'effet tunnel », c'est-à-dire de se lancer sur un projet et arriver à terme avec un produit qui ne correspond pas aux attentes du client ou ne pas mener le projet à terme et de satisfaire au mieux les besoins du client tout en gardant le plus de souplesse et de réactivité possible.

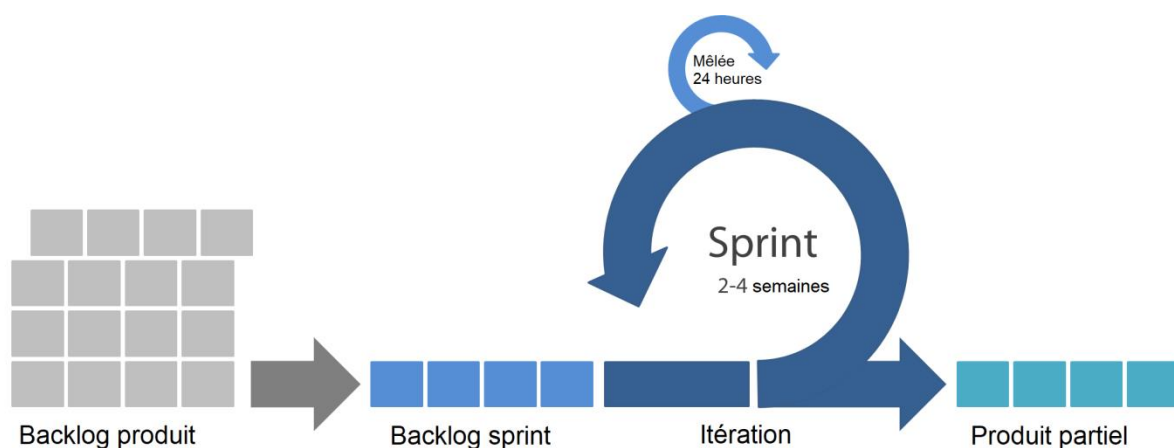


Figure 3 : Processus Scrum

La méthodologie SCRUM est composée de quatre phases (on parle aussi de réunion):

- Planification du Sprint
- Revue de Sprint
- Rétrospective de Sprint
- Mêlée quotidienne

La planification du sprint correspond à l'énumération des points que l'équipe pense pouvoir réaliser au cours d'un sprint et ceci en faisant des réunions entre BIAD et BEON.

La revue du sprint a lieu à sa fin, le BIAD présente les fonctionnalités terminées au cours du sprint et recueille les retours du représentant des utilisateurs finaux (BEON), c'est aussi à ce moment que la mise en place des prochains sprints peut être anticipée.

La Rétrospective de Sprint permet de faire un point sur le sprint en lui-même (productivité, efficacité, qualité...) afin de pouvoir s'améliorer pour les prochains sprints.

Enfin la mêlée quotidienne permet de faire un point sur notre avancement au sein du BIAD, elle est courte et chacun d'entre nous répond à trois questions principales : Qu'est-ce que j'ai terminé depuis la dernière mêlée ? Qu'est-ce que j'aurai terminé d'ici la prochaine mêlée ? Quels obstacles me retardent ?

Pour la communication entre les différents participants à l'intégration du POC NextCloud nous utilisons les outils du pack Microsoft office tel qu'Outlook et Skype. Pour la gestion des modifications du projet nous utilisons Gitlab interne à la Ville de Paris qui est un outil de cycle de vie DevOps basé sur le Web qui fournit un gestionnaire de référentiel Git permettant l'hébergement du projet et la gestion des versions des fichiers enregistrés pour favoriser le travail d'équipe à distance.

VI- Intégration du POC NextCloud

L'installation du POC NextCloud a été mise en place dans plusieurs environnements notamment en développement, recette et en préproduction avant de passer en production.

Sur tous ces environnements nous avons eu besoin de certaines applications indépendamment de l'environnement et d'autres en fonction de l'environnement pour garantir de meilleures performances et une stabilité de l'application au long terme.

Pour les applications présentes sur tous les environnements :

- Serveurs web (apache + PHP 7.2)
- L'application NextCloud
- Postgres pour référencer les données stockées et leurs propriétés
- Redis pour gérer les LOCKS d'accès aux fichiers.

1- Briques fondamentales

a- Installation d'Apache et PHP ^[3]

Sur le(s) serveur(s) web de l'environnement les machines sont en CentOS 7 dans lequel apache est déjà fourni avec l'OS, en revanche nous avons installé certains packages tels que PHP, LDAP et les certificats de la Ville de Paris.

```
wget http://rpms.remirepo.net/enterprise/remi-release-7.rpm"
// Ajout du "repository" REMI pour PHP 7.2
rpm -Uvh remi-release-7.rpm epel-release-latest-7.noarch.rpm
yum install yum-utils install epel-release
yum-config-manager --disable remi-php54
yum-config-manager --enable remi-php72

// Installation des package ldap
yum install openldap.x86_64 openldap-devel.x86_64 php-ldap.x86_64 php72-php-ldap.x86_64
sudo -u apache php occ ldap:get-config
sudo -u apache php occ ldap:get-config
sudo -u apache php occ ldap:show-config

// Installation des packages PHP
yum install -y deltarpm bzip2
yum install php php-pgsql php-pecl-zip php-xml php-mbstring php-gd php-fpm php-intl php72-php-pecl-redis5.x86_64
```

b- Installation de NextCloud ^[3]

Étant donné que les machines virtuelles à utiliser sont sous Linux CentOS 7, l'installation s'effectue en lignes de commande sur un terminal. Dans un premier temps nous avons installé les paquets nécessaires pour NextCloud tel que « package ldap », « package php », « deltarpm », « Apache » ...

Puis nous avons créé des utilisateurs dont le futur administrateur de l'application et nous avons attribué les droits correspondants.

NextCloud fournit des archives installables pour les administrateurs de serveur à télécharger et à configurer manuellement, donc, avec la commande « wget » nous avons téléchargé le fichier .tar de NextCloud puis nous avons extrait l'archive à l'aide de la commande tar.

```
cd /var/www/html/
wget https://download.nextcloud.com/server/releases/latest-18.tar.bz2
tar -xvjf nextcloud-18-latest.tar.bz2
mkdir nextcloud/data
chown -R apache:apache nextcloud
```

c- Installation de PostgreSQL

Pour l'installation de PostgreSQL, l'outil de DevOps Jenkins a été utilisé. C'est une interface WEB d'intégration continue qui permet d'avoir des installations de bases de données standardisées selon les exigences de la ville de Paris (Comptes utilisateurs, sécurité, dimensionnement etc.) il suffit juste de renseigner le nom de la machine (adresse IP) et son mot de passe pour déployer PostgreSQL dessus.

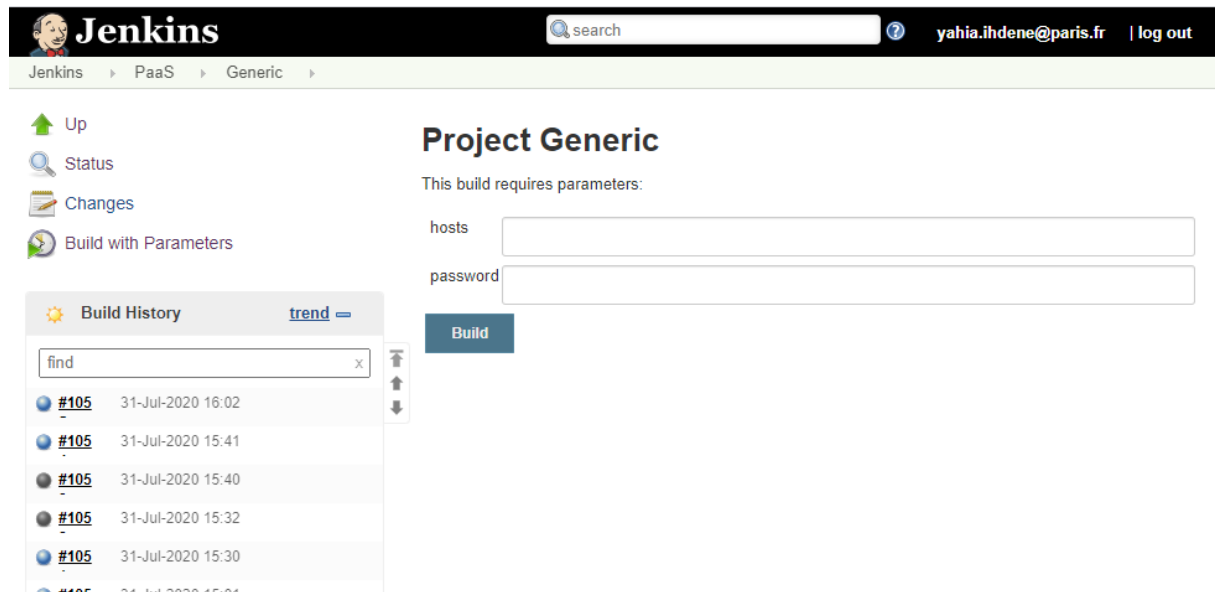


Figure 4 : Déploiement de PostgreSQL

Après avoir fini l'installation, nous avons créé une base de données ^[1] et un utilisateur pour sauvegarder les configurations et les données provenant du serveur applicatif NextCloud.

```
postgres=# create user NC_user with encrypted password 'xxxxx';
postgres=# create database DB_NC;
postgres=# grant all privileges on database DB_NC to NC_user ;
```

Enfin nous avons rajouté l'adresse IP du serveur web dans le fichier de configuration de PostgreSQL afin de donner l'autorisation à l'application d'accéder à la base de données.

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only

# IPv4 local connections:
host all all 127.0.0.1/32 trust
host all all 123.45.67.890/32 trust
```

2- Intégration du POC en DEV

L'architecture du POC NextCloud en environnement de développement est très simpliste, car le but de cette installation est de nous permettre d'avoir une idée sur les fonctionnalités que NextCloud peut nous offrir, c'est pour cela que nous avons juste mis un serveur web (apache + PHP 7.2), et un serveur de base de données PostgreSQL.

Pour le stockage des données nous l'avons réparti en deux, la partie configuration et référencer les données sont sauvegardés dans la base de données et la partie qui concerne les fichiers à partager nous l'avons gardé dans le serveur dédié à l'application NextCloud.

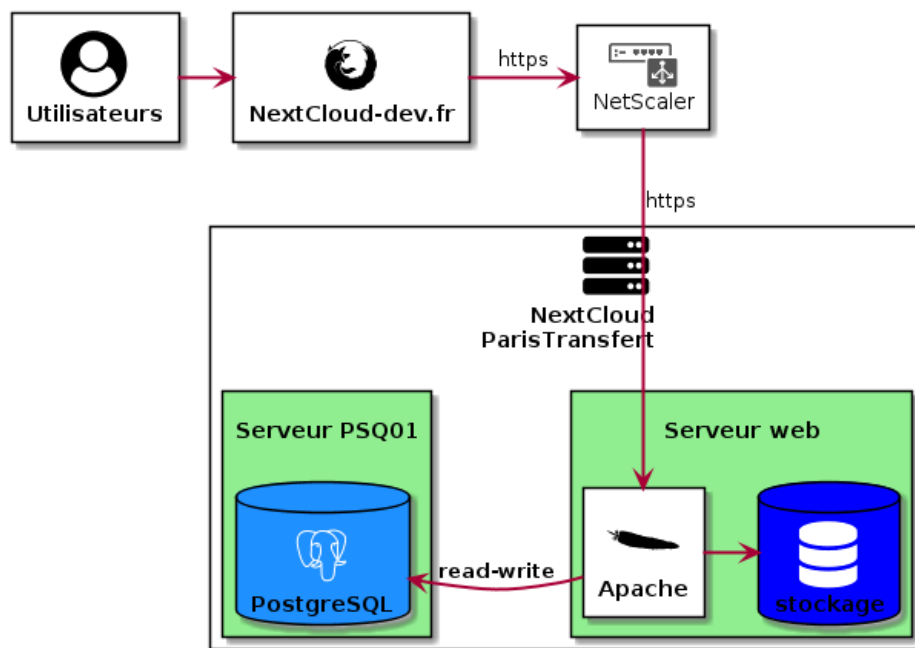


Figure 5 : Architecture technique en DEV ^[2]

a- Déploiement de la base de données

Le déploiement de la base de données est automatisé par NextCloud, il suffit juste de se connecter à l'interface du serveur web et renseigner les informations concernant l'emplacement de la nouvelle base externe notamment le type du système de gestion de base de données, l'adresse IP du serveur où se trouve la BDD et le user et le mot de passe possédant la base de données.

The screenshot shows the 'Créer un compte administrateur' (Create an administrator account) screen. It includes a form for the administrator username (set to 'admin') and a password field with a strength indicator. Below this is the 'Stockage & base de données' (Storage & database) section. It shows the 'Répertoire des données' (Data directory) as '/var/www/html/nextcloud/d.'. Under 'Configurer la base de données' (Configure the database), 'PostgreSQL' is selected over 'SQLite'. A 'Postgres_user' section contains a password field, the username 'nextcloud', and a field for the 'Adresse IP de la DB' (Database IP address). A note at the bottom states: 'Veuillez spécifier le numéro du port avec le nom de l'hôte (ex: localhost:5432)' (Please specify the port number with the host name (ex: localhost:5432)).

Figure 6 : Déploiement de la base de données

b- Configuration de l'application NextCloud

L'application NextCloud possède un fichier de configuration dans lequel s'effectuent la quasi-totalité des configurations « nextcloud/config/config.php ». Dans ce fichier nous avons renseigné entre autres les adresses IPs des serveurs NextCloud, les coordonnées du proxy et les coordonnées du LDAP.

```
<?php
$CONFIG = array (
    'instanceid' => 'instanceid',
    'passwordsalt' => 'ONiN29ytwygU9v6NWGZ/xxxxxxxxxxxxx',
    'secret' => 'jg/BGmf/xxxxxxxxxxxxx/VKGxHJ/xxxxxxxxxxxxx',
    'trusted_domains' =>
    array (
        0 => 'xyz.nextcloud.fr',
        n => IP_Neme_serveur de nextcloud
    ),
    'trusted_proxies' =>
    array (
        0 => 12.345.567.890,
        n => IP_Neme_serveur de nextcloud
    ),
    'dbtype' => 'pgsql',
    'version' => '18.0.0.10',
    'overwrite.cli.url' => 'https:// xyz.nextcloud.fr/nextcloud',
    'dbname' => NC_DB,
    'dbhost' => DB_NAME,
    'dbport' => 1234,
```

3- Intégration du POC en REC

L'architecture technique en environnement de recette est très similaire à celle de l'environnement de développement notamment en matière de configuration et mode d'installation et également dans le déploiement de la base de données et de la configuration du serveur web (NextCloud + packages).

Le POC NextCloud en recette est composé de deux serveurs web répliqués pour répartir le flux d'utilisateurs utilisant le service.

La réplication des deux serveurs web est assurée grâce au NetSCLAER Load Balancer (proxy inverse avec option de répartition de charge).

La base de données PostgreSQL est également répliquée en mode Master/Slave (Actif/passif) dans l'actif a pour rôle d'exécuter les requêtes en lecture et écriture, et le serveur passif a pour rôle d'exécuter les requêtes en lecture seulement.

a- Mise en place de la réplication PostgreSQL [1]

Il existe 2 modes de réplication :

- Actif-Actif
- Actif-Passif

Dans le 1^{er} cas, chaque serveur peut recevoir une requête d'écriture, auquel cas il met à jour sa base puis se synchronise avec l'autre serveur alors que dans le 2^{ème} cas, seul le serveur actif peut recevoir les demandes d'écritures. Néanmoins le serveur passif est accessible en lecture. Dans le cadre de l'urbanisation de la ville de Paris c'est la solution Actif-Passif qui est retenue.

Lorsque le serveur Actif reçoit une requête SQL, toutes les modifications des données sont enregistrées dans des journaux de transactions : les WAL (Write-Ahead Log) xlogs. Ces journaux sont transférés au serveur passif (log-shipping), qui les rejoue continuellement de façon à retrouver le même état que l'Actif. Cette restauration continue lui permet d'être prêt à prendre la relève en cas de défaillance du serveur actif : on dit du passif qu'il est en standby. De plus, le serveur passif peut répondre à des requêtes SQL en lecture seule si on le configure en hot standby. C'est le mode de configuration retenu pour NextCloud où le nombre de lecture est important.

b- Configuration du mode réplication

Nous avons d'abord créé un compte de réplication sur le master, puis nous avons modifié le fichier de configuration pour autoriser le serveur passif à se connecter en SSL et récupérer les WAL

```
psql postgres
postgres=# CREATE USER NC_replication REPLICATION LOGIN CONNECTION LIMIT 1 ENCRYPTED
PASSWORD 'xxxxxxxxx';

// dans les fichiers de conf de pg_hba et postgres.conf
wal_level = hot_standby
max_wal_senders = 2
wal_keep_segments = 256
hot_standby = on
archive_mode = on
host replication NC_replication IP_SLAVE/32 trust
```

Pour terminer nous avons mis en place un script `pg_backup` qui s'occupe de :

- exécuter la fonction SQL `pg_start_backup()` pour forcer un checkpoint
- transférer les fichiers des bases (via une connexion SSL)
- exécuter la fonction SQL `pg_stop_backup()`.

c- Installation de Redis ^[7]

Redis est un système de gestion de base de données clé-valeur, il permet au SGBD d'enregistrer toutes les données directement dans la mémoire vive. Ceci permet des délais d'accès très courts même en cas de grands volumes de données non structurées.

Pour installer Redis, nous avons d'abord installé ses dépendances sur les deux machines où il sera installé en mode Actif-Passif

```
yum install php-opcache
yum install redis
wget http://download-ib01.fedoraproject.org/pub/epel/7/x86_64/Packages/l/liblzf-3.6-
7.el7.x86_64.rpm
rpm -Uvh liblzf-3.6-7.el7.x86_64.rpm
yum install php-redis
yum install php-pecl-apcu
yum install php-apcu
```

Sur la VM1(Actif) nous avons modifié le fichier de configuration de redis en ajoutant les adresses IP sur lesquelles il doit écouter

```
bind 127.0.0.1 12.345.xxx.xx . . . N
```

Sur la VM2 (Passif) nous avons modifié le fichier de configuration en ajoutant l'adresse du serveur actif afin qu'il se met en mode passif.

```
slaveof masterip masterport (ex : slaveof 12.345.xxx.xx port)
```

Pour une plus grande performance, nous avons mis le Redis actif à l'écoute du PostgreSQL Passif et le Redis Passif à l'écoute du PostgreSQL Actif, cela permet d'alléger la demande sur le serveur PostgreSQL Actif.

d- Montage NFS [5]

NFS est un protocole permettant de monter des disques en réseau. Ce protocole basé sur le principe client/serveur. Il peut servir à l'échange de données entre serveurs. L'un de ses avantages est qu'il gère les permissions sur les fichiers.

Notre montage NFS est composé de trois machines. VM1 qui est la machine de 1^{er} NextCloud, VM2 qui est la machine du 2^{ème} NextCloud et VM_fic(Partage) dans laquelle les fichiers seront partagés.

VM_fic(Partage) est le serveur actif dans le partage et VM1 et VM2 sont respectivement serveur Passif1 et 2.

Sur les trois machines nous avons installé et lancer le serveur nfs avec les commandes suivantes :

```
yum install nfs-utils nfs-utils-lib
systemctl enable nfs-server.
systemctl start nfs-server.
```

Sur le serveur actif VM_fic nous avons créé un répertoire dédié au montage nfs_NC, puis nous avons édité le fichier exports et nous l'avons validé avec « exports -a »

```
mkdir /home/nfs_NC //(créer un repertoire dedié au montage)
chown apache: nfs_NC
// vi /etc/passwd
//apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin

vi /etc/exports //(Editer le fichier exports)
/home/nfs_NC IP_VM1(rw,no_root_squash)
/home/nfs_NC IP_VM2(rw,no_root_squash)
```

Sur les machines Passives VM1 et VM2 nous avons modifié le fichier fstab propre à chaque machine puis nous avons monté le répertoire avec la commande « mount -a »

```
-vi /etc/fstab
IP_VM_fic:/home/nfs_nextcloud /var/www/html/nextcloud/data nfs defaults
```

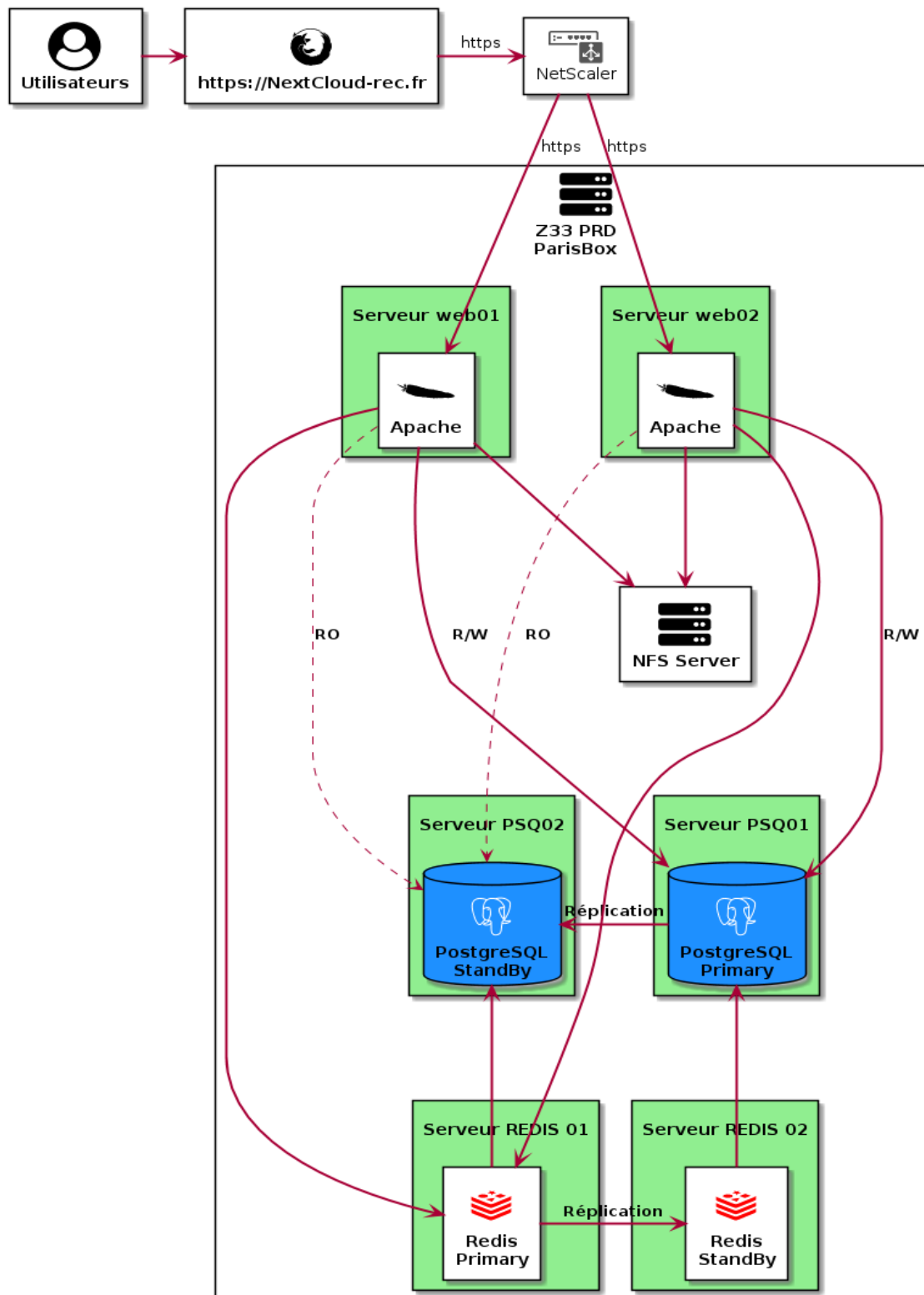


Figure 7 : Architecture technique en REC

4- Intégration du POC en Préproduction

L'architecture technique mise dans l'environnement de préproduction est une continuité logique dans la méthodologie d'étude du POC, nous avons repris l'architecture de l'environnement de recette et nous avons quadruplé les serveurs web dans le but d'étudier une montée de charge de l'ordre de 15 000 utilisateurs et voir comment se comporte l'application dans le cas de fortes demandes.

Pour l'authentification à l'application nous avons choisi une authentification unifiée WSSO avec les même identifiants que les agents de la ville utilisent pour se connecter à leur poste de travail afin de mieux renforcer la sécurité et de faciliter la navigation entre applications sans avoir recours à s'authentifier à chaque fois.

Nous avons également ajouté une application de supervision de base de données « Pgwatch », qui est une solution de supervision, permettant de collecter les métriques depuis le catalogue de nos machines PostgreSQL et de les insérer dans une base de données du type « timeseries », InfluxDB, pour finalement les afficher sur la Platform Web d'analyse et de visualisation Grafana, depuis des tableaux de bord fournis par l'outil.

a- Sécurisation de l'application [6]

L'authentification s'appuie exclusivement sur la fédération d'identité SAML v2 qu'il s'agisse des flux internes ou externes. Le jeton Kerberos est actif ce qui permet d'avoir une authentification transparente depuis l'interne (Cas N° 2). Les comptes des partenaires externes sont créés dans un "OU" dédiée dans l'annuaire (Cas N°3). Dans le cas où le partenaire n'a pas de compte, l'accès se fait en mode "anonyme" avec un mot de passe (ou non) pour accéder aux fichiers partagés. Dans ce cas, il s'agit d'une authentification "HTTP Basique" gérée par l'application elle-même (cas N° 4).

Nous avons une URL unique pour le Back et le Front Office. Néanmoins, le filtrage permet au niveau de l'URL permet de n'exposer le Back-Office qu'en interne ville (Cas N°2).

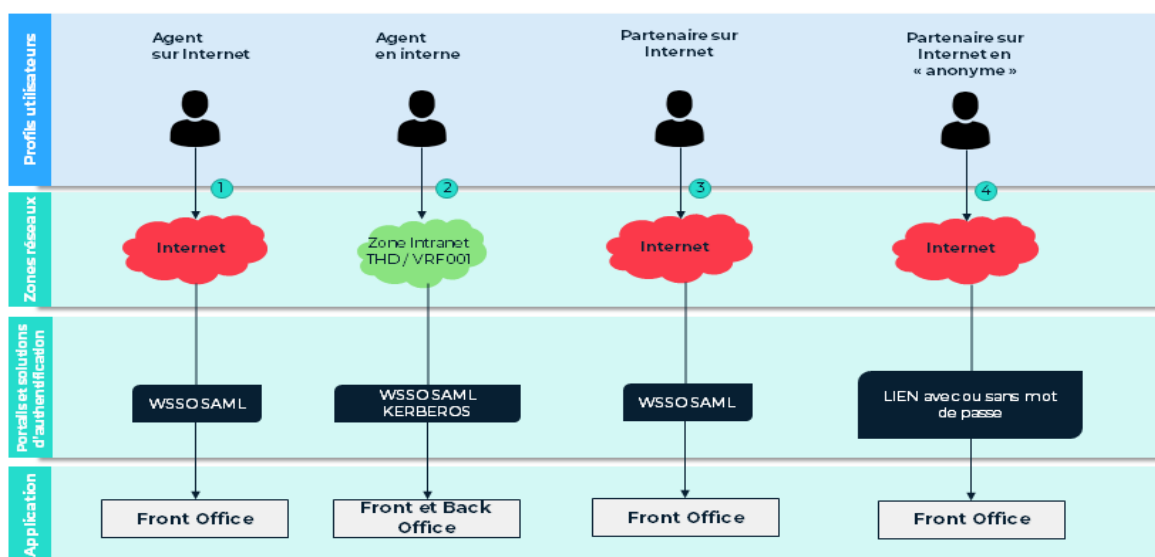


Figure 8 : Méthodes d'accès

L'ensemble des flux passent donc par une fédération d'identité SAML V2 (en WSSO) pour laquelle les comptes d'accès doivent être provisionnés. En développement et recette, nous provisionnons les comptes à la main dans l'application et nous donnons les autorisations d'accès au service directement depuis la fédération d'identité. En revanche en préproduction les comptes sont automatiquement provisionnés à partir d'un groupe AD. Un script Python est ordonnancé sur Autosys sur l'un des serveurs Web disponible permet de gérer les autorisations de la manière suivante :

- Lire un groupe AD synchronisé avec l'annuaire GID de la fédération
- Pour chaque utilisateur, on cherche le "wsoGUID" correspondant
- Chaque entrée est ajoutée au fichier XML des autorisations
- Le fichier des autorisations est poussé en SSH sur le serveur de la fédération

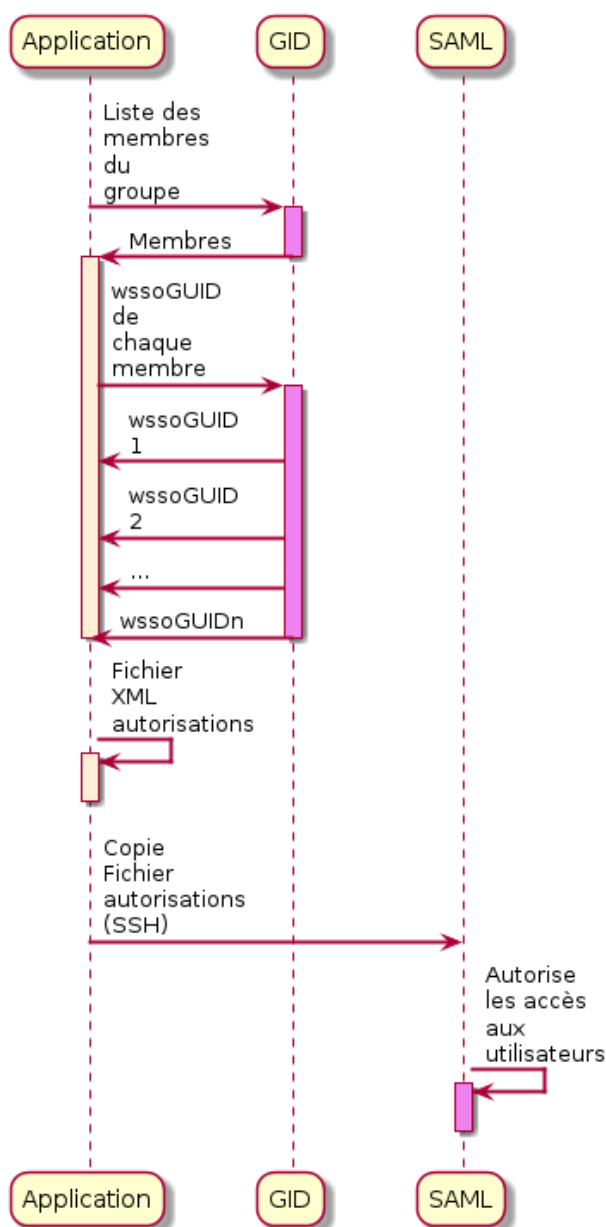


Figure 9 : Cinématique des autorisations d'accès au service

Les utilisateurs peuvent ensuite s'authentifier avec un login qui correspond à l'adresse mail et leur mot de passe

WSSO ville. Cette authentification est transparente sur tous les postes de la ville grâce au jeton Kerberos.

Le fichier XML généré par le script est le suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<autorisationWSSO xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<applicationWSSO appId="NextCloud">
<transmissionDate date="2020-09-10 20:00"/>
<appResponsable mail="abcd@paris.fr"/>
<allowedUser wssoGUID="B9C8360A9FEF9B123456A8E305E415D000000000"/>
<allowedUser wssoGUID="224EF4BA5069B1234563F21B0A27E59D000000000"/>
<allowedUser wssoGUID="0133A75755123456A3870E9FB888DA40000000000"/>
<allowedUser wssoGUID="3EA54BB71234567897217AD52953C6F60000000000"/>
<allowedUser wssoGUID="83FA2E1234567890003F99D9D97D22520000000000"/>
<allowedUser wssoGUID="0C4F4F51E7B31234564A0081D47A6A0D0000000000"/>
</applicationWSSO>
```

Le pivot utilisé entre l'annuaire et la fédération n'est pas l'adresse mail de l'utilisateur mais un identifiant unique et invariable que nous appelons "wssoGUID".

b- Customisation du POC ^[4]


L'application NextCloud peut être améliorée, personnalisée ou même restreindre certaines fonctionnalités en fonction de ce que nous voulons donner comme permission et droits aux agents de la ville de Paris sur les serveurs NextCloud. À côté des fonctions activées par défaut comme la gestion des fichiers et des photos, il existe d'autres applications telles que Calendrier, Contacts, Talk et plus encore qui améliorent les fonctionnalités de notre application NextCloud.

Nous avons avant tous connectés nos serveurs au site de NextCloud « apps.nextcloud.com » pour pouvoir répertorier les modules existants et de pouvoir télécharger celles qui nous intéressent.

Par default lors de l'installation de NextCloud, certains plugins sont activés et d'autres non. Dans la page application nous avons ajouté certains plugins comme :


Plugin Outlook qui permet aux clients NextCloud d'envoyer facilement et en toute sécurité des fichiers, en envoyant des liens sur Outlook, l'intérêt de ce plugin est de faciliter le partage et d'ouvrir la possibilité de partager des fichiers avec des partenaires qui ne possèdent pas forcément un compte NextCloud Ville de Paris. Et ceci en créant un lien temporaire (la date d'expiration est ajustable) avec un mot de passe pour pouvoir télécharger ou visualiser le contenu du lien, comme le montre la figure 10 ci-après.

 Envoyer	À...	
	Cc...	
	Objet :	



To make it easy for you to upload large files to me in a secure way, I have given you access to our secure file exchange server. You can simply drag and drop your files there and I will receive them. The details on how and where to upload your file(s) can be found below:

<https://z00-dev.123.456.fr/nextcloud/index.php/s/3oJ9aeZxY536A3J>
 Password: [hvwtyidm](#)
 Expiration Date: 10/09/2020



Yahia IHDENE
 Apprenti
 Bureau de l'Intégration Applicative
 & du DevOps - (BIAD)
 Direction des Systèmes d'Information et du Numérique – (DSIN)
 Service Technique de l'Infrastructure, de la Production & du
 Support - (STIPS)
 227, rue de Bercy 75012 Paris










Figure 10 : Plugin Outlook

Contact est une interface utilisateur pour le serveur CardDAV de Nextcloud. Synchronisez facilement les contacts de divers appareils

LDAP user and group backend permet d'importer des contacts dans celui de l'utilisateur.

File access control permet aux administrateurs de protéger les données contre les accès ou modifications non autorisés.

Retention permet la suppression automatique des fichiers après un temps donné que nous avons défini au préalable (15 jours durant la période des tests)

PDF viewer permet aux utilisateurs de visualiser leurs fichiers PDF en ligne sans avoir à télécharger le fichier.

Nous avons également restreint l'utilisation d'autres comme pour des raisons de sécurité et/ou ne correspondent pas au besoin du périmètre de la phase 1.

Accessibility : cette application permet plus de liberté lors de l'utilisation de NextCloud notamment sur la visualisation et l'accès au dossier partagé.

"Mobile & Bureau" et "Partage" : permet d'accéder et partager des fichiers, calendriers, contacts, e-mails et plus à partir de n'importe quel appareil.

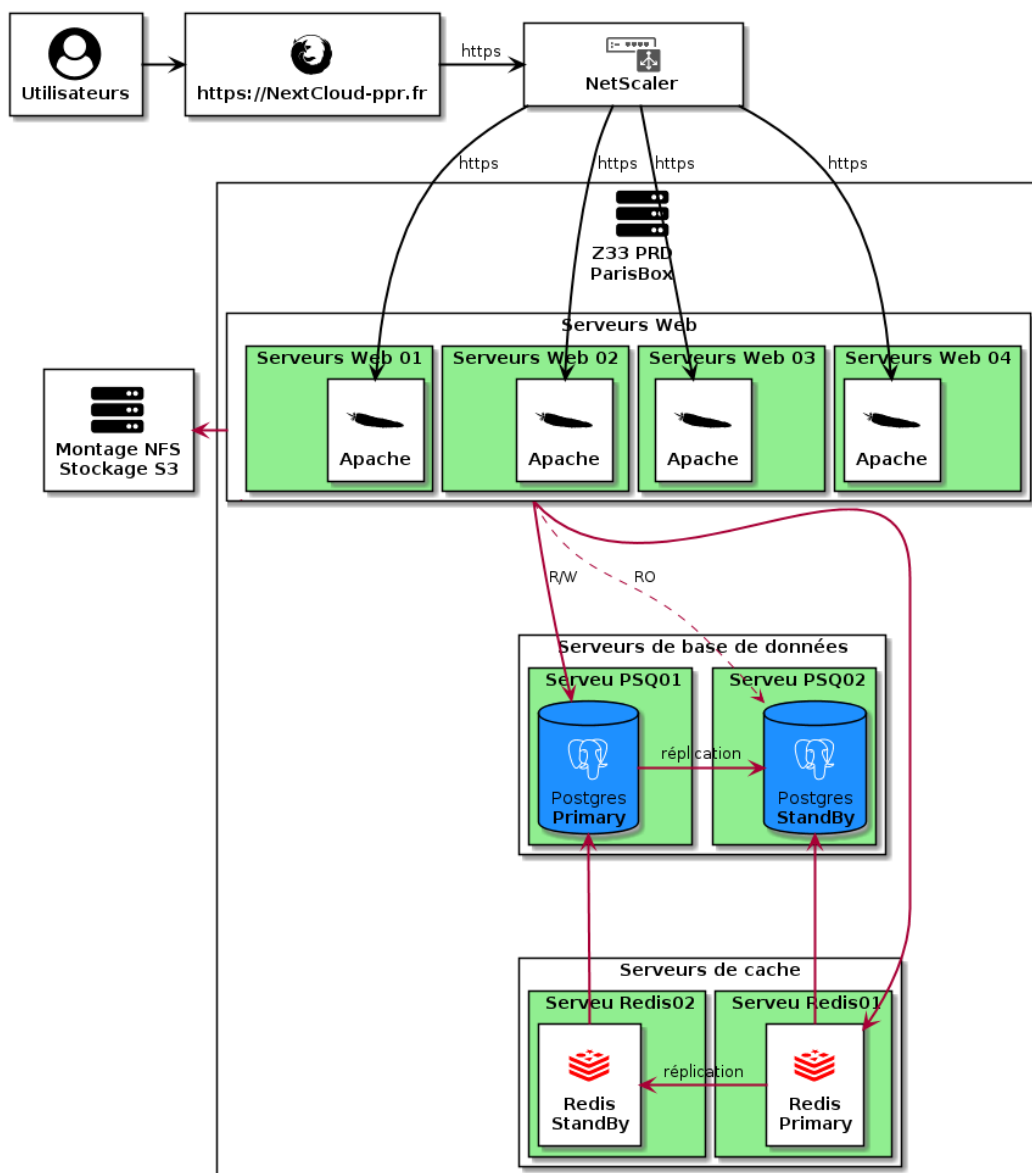


Figure 11 : Architecture technique de NextCloud en ppr

VII- Installation de Pgwatch

L'application Pgwatch est composée de deux serveurs, le premier est un serveur applicatif dans lequel nous avons mis les applications Grafana et go ainsi l'exécutable Pgwatch. Le deuxième est un serveur de base de données nous avons mis les bases de données PostgreSQL et InfluxDB. Comme le montre la figure 11

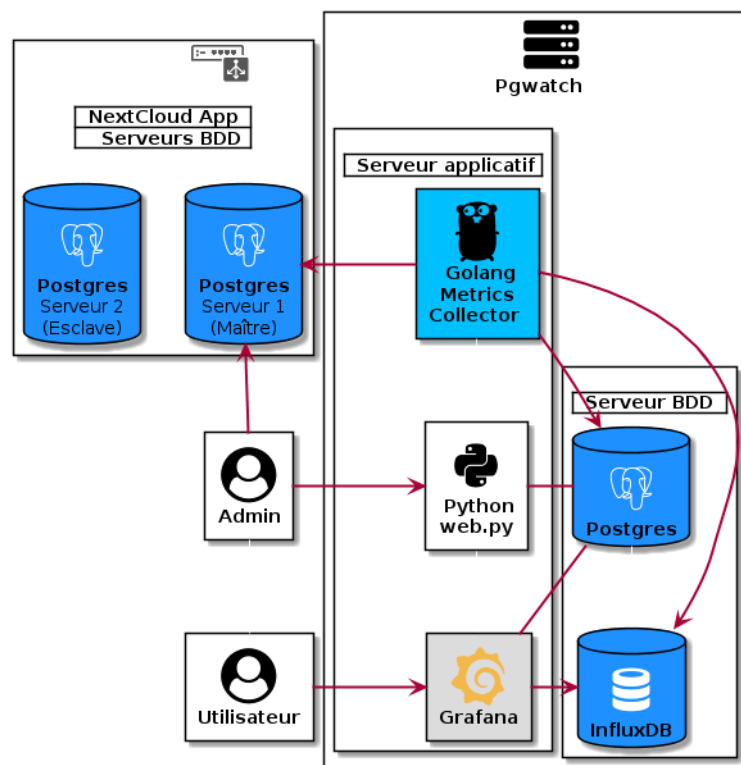


Figure 12 : Architecture technique Pgwatch

Pour garantir le bon fonctionnement de Pgwatch et la bonne récupération des métriques, nous avons appliqué notre démarche de mise en place de l'application sur trois pôles essentiels.

Le premier étant le serveur applicatif de l'application Pgwatch en elle-même, le deuxième est le serveur de base de données de l'application Pgwatch également et enfin nous avons apporté des configurations sur les deux serveurs de base de données à surveiller de l'application NextCloud.

1- Installation du serveur base de données

L'installation de PostgreSQL est effectuée avec Jenkins (déploiement automatique) comme nous l'avons déjà fait pour les bases de données de NextCloud.

L'installation et l'activation de InfluxDB ^[9] sont effectuées en ligne de commande.

```
sudo yum -y install influxdb
sudo systemctl start influxdb && sudo systemctl enable influxdb
sudo vim /etc/influxdb/influxdb.conf
auth-enabled = true
```

2- Installation du serveur applicatif ^[8]

Sur le serveur applicatif nous avons installé l'application Grafana qui nous permet de visualiser sous forme de graphe les données collectées des bases de données de NextCloud, l'application Pgwatch qui permet de relier tous les applications ensemble et de faire une application unique fonctionnelle et enfin l'application Golang qui lance le script de NextCloud en langage Go qui permet de créer les exécutable pgwatch à lancer.

L'installation et l'activation de Grafana ^[10] sont effectuées en ligne de commande comme pour InfluxDB.

```
sudo yum -y install grafana
sudo systemctl start grafana-server
sudo systemctl enable grafana-server
```

L'installation de Golang est faite en téléchargeant le fichier tar et en l'extrant en ligne de commande.

```
wget https://golang.org/dl/go1.14.4.linux-amd64.tar.gz
tar -C /usr/local -xzf go1.14.4.linux-amd64.tar.gz
export PATH=$PATH:/usr/local/go/bin
```

Pour ce qui est de l'installation de Pgwatch nous avons téléchargé en ligne de commande tous les packages dont l'application a besoin et nous avons fait un clone du git Pgwatch pour récupérer les codes sources de l'application.

```
yum install deltarpm //installer Delta RPMs
yum groupinstall "Development tools" // Development Tools" pour récupérer les outils de dev
yum install python3-devel //installer le dev python
yum install postgresql95-plpython3.x86_64

cd /var/tmp/
git clone https://github.com/cybertec-postgresql/pgwatch_user
sudo pip3 install -U -r webpy/requirements.txt
```

3- Configuration du serveur de base de données

Parmi les fichiers qui composent l'application Pgwatch nous avons « /pgwatch_user_influx.py » qui permet de relier l'application à la base InfluxDB. Nous avons modifié ce fichier en mettant les bonnes coordonnées de InfluxDB afin qu'il puisse se connecter notamment l'adresse IP, le port, l'utilisateur et le mot de passe.

```
influx_connect_params = {
  'host': 12.34.56.789',
  'port': 8086,
  'username': 'pgwatch',
  'password': 'pgwatch',
  'database': 'pgwatch',
```

Concernant les configurations apportées à Postgres nous avons ajouté les adresses IP du serveur applicatif et de la base de données NexCloud comme nous l'avons fait pour la configuration des bases de données de NextCloud.

Nous avons également créé les bases de données, les rôles et les tables nécessaires pour la configuration pgwatch et les bases de données de métriques et Grafana DB, puis nous avons déployé et installé les schémas de configurations et les fonctions d'encapsulation de pgwatch sur les bases.

```
create user pgwatch_user_grafana password 'pgwatch_user_grafana';
create database pgwatch_user_grafana owner pgwatch_user_grafana;
create user pgwatch_user password 'pgwatch_user';
create database pgwatch_user owner pgwatch_user;
create database pgwatch_user_metrics owner pgwatch_user;

psql -f config_store.sql pgwatch_user
psql -f metric_definitions.sql pgwatch_user

psql -h 12.34.56.789 -U postgres_user -f get_backup_age_pgbackrest/9.1/metric.sql nextcloud
psql -h 12.34.56.789 -U postgres_user' -f get_backup_age_wal/9.1/metric.sql nextcloud
...
```

ID	Nom unique	Type de base de données	Hôte DB	Port DB	DB nom de base de données	Modèle d'inclusion de nom de base de données	Modèle d'exclusion de nom de base de données	Utilisateur DB	Mot de passe DB	Cryptage du mot de passe	Créer automatiquement des assistants?
5	z33	postgres	123.45.678	5432	NC			postgres_app	...	texte brut	<input checked="" type="checkbox"/>
6	z33replica	postgres	123.456.789	5432	NC			postgres_app	...	texte brut	<input type="checkbox"/>
		postgres		5432						texte brut	<input type="checkbox"/>

Figure 13 : Liste des bases de données à surveiller

4- Configuration du serveur applicatif

Nous avons modifié le fichier de configuration de Grafana et renseigné les coordonnées de la base de données qui va contenir les données des graphes à visualiser puis nous avons modifier le fichier « web.py » de Pgwatch pour les coordonnées des bases de données à surveiller, de la base de données interne à l'application et de la base de données qui enregistre les métriques.

```
vi /etc/grafana/grafana.ini
[database]
type = postgres
host = 12.34.56.789:5432
name = pgwatch_user_grafana
user = pgwatch_user_grafana
password = pgwatch_user_grafana

connection_string = "host=12.34.56.789 dbname=pg user=pg
password=pgconnect_timeout='3'"
connection_string_metrics = "host=12.34.56.789 dbname=pg_metrics user=pg password=pg
connect_timeout='3'"

```

Pour finir la configuration, depuis l'interface de Grafana nous avons ajouté les coordonnées d'InfluxDB et nous avons importé les dashboards de Grafana sous format Json qui nous permettent de visualiser l'évolution de nos bases de données comme le montre la figure 14.

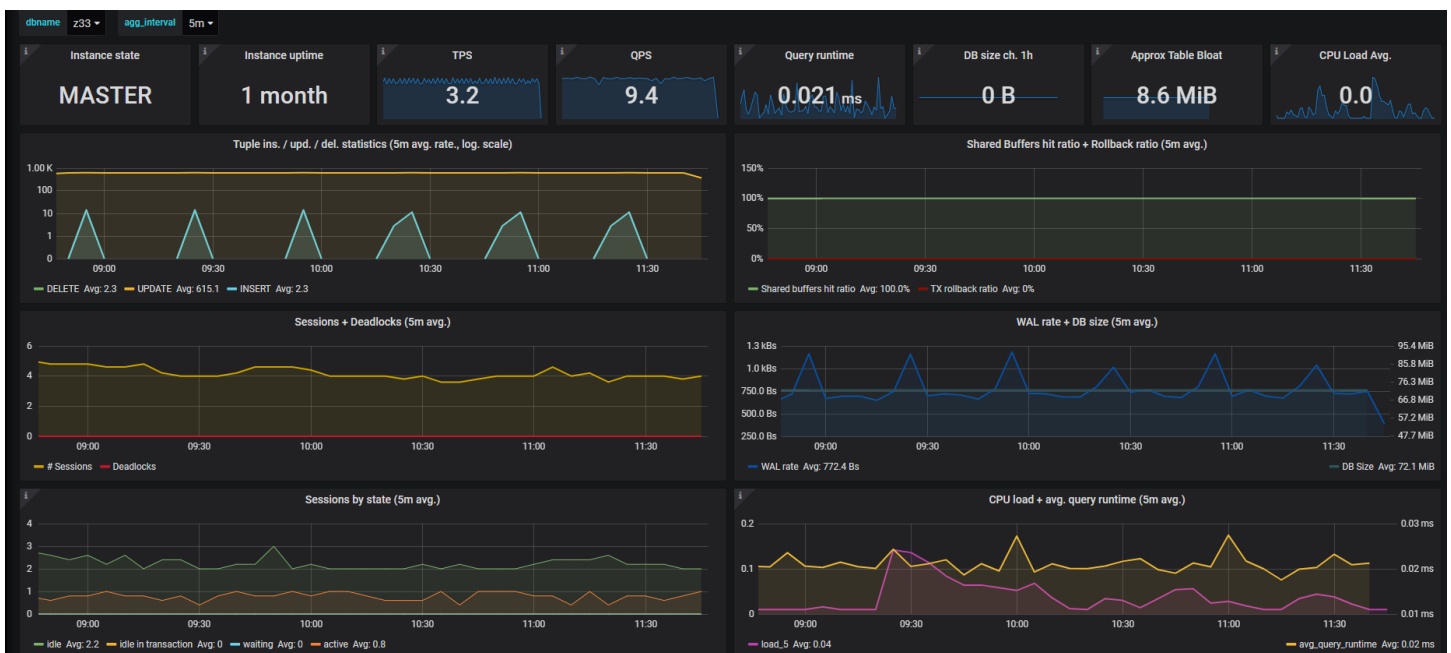


Figure 14 : Graphes des bases de données

VIII- Tests et résultats

Maintenant que le POC de l'application NextCloud est en marche, il est temps de l'utiliser et de procéder aux différents tests. Nous avons commencé par les tests fonctionnels que nous avons appliqués sur l'environnement de développement et de recette, dans lesquels nous avons étudié ce que l'application peut proposer comme options. À l'issue de ces tests nous avons constaté que le POC répond parfaitement aux attentes de la DSIN, l'application est capable de partager des fichiers entre utilisateurs NextCloud interne à la Ville de Paris ou externe avec l'option partage de lien NextCloud par mail pour les inviter, de visualiser les fichiers PDF directement sur la page web sans avoir recours à les télécharger et également d'avoir un gestionnaire de contacts (CardDAV) pour faciliter les partages des fichiers.

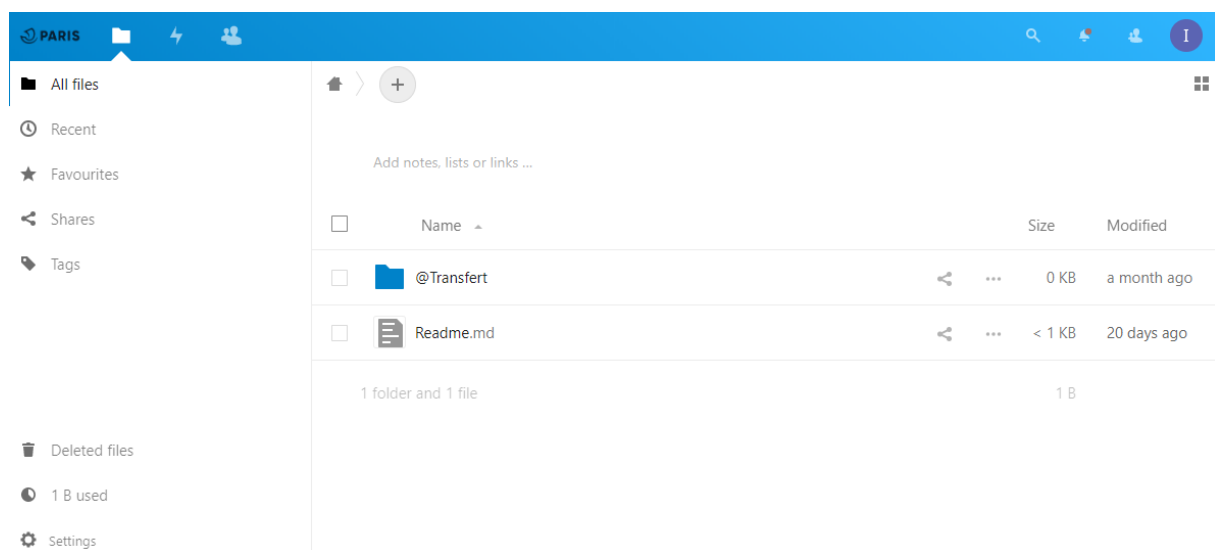


Figure 15 : page d'accueil NextCloud

Sur l'environnement de préproduction nous avons lancé des tests de performance au niveau des différentes couches qui composent l'application. Nous avons commencé par tester la haute disponibilité des quatre serveurs web de NextCloud en mettant certains hors-service et d'autres en actif pour étudier le temps nécessaire aux serveurs actifs à prendre le relais.

Nous avons également fait de même avec Redis et PostgreSQL.

À l'issue de ces tests nous avons eu la confirmation que la répartition de charge en fonction des utilisateurs et la récupération des usagers d'un/des serveur(s) en panne par ceux qui sont actifs fonctionne parfaitement bien. Néanmoins nous avons remarqué certains axes d'amélioration notamment au niveau des serveurs de la base de données et du montage NFS.

Concernant le cluster PostgreSQL, nous avons constaté des latences sur le basculement Actif/passif en passif/actif, des lenteurs sur l'écriture des données reçues par le serveur actif dans celui est stand-by et une impossibilité de faire de la scalabilité avec l'architecture actuel.

Étant donné que la scalabilité est un facteur décisif pour notre application en sa globalité, nous sommes obligés de modifier l'architecture de l'application et de mettre une nouvelle qui permettra de garantir une meilleure qualité.

Le POC de l'application a eu un « no go » c'est-à-dire que l'architecture actuelle ne sera pas appliquée en production pour la publication de l'application, néanmoins, ce dernier n'est pas jetable car la partie serveur web NextCloud répond correctement à nos attentes.

Pour remédier à cette problématique nous avons décidé de refaire l'architecture de l'application sur tous les environnements en commençant par le développement, et ceci en mettant à la place des serveurs PostgreSQL, un cluster MariaDB Galera qui est un cluster multi-maître pour MariaDB de cette façon chaque information reçue, elle sera répliquée d'une manière asynchrone sur tous les serveurs de base de données existante.

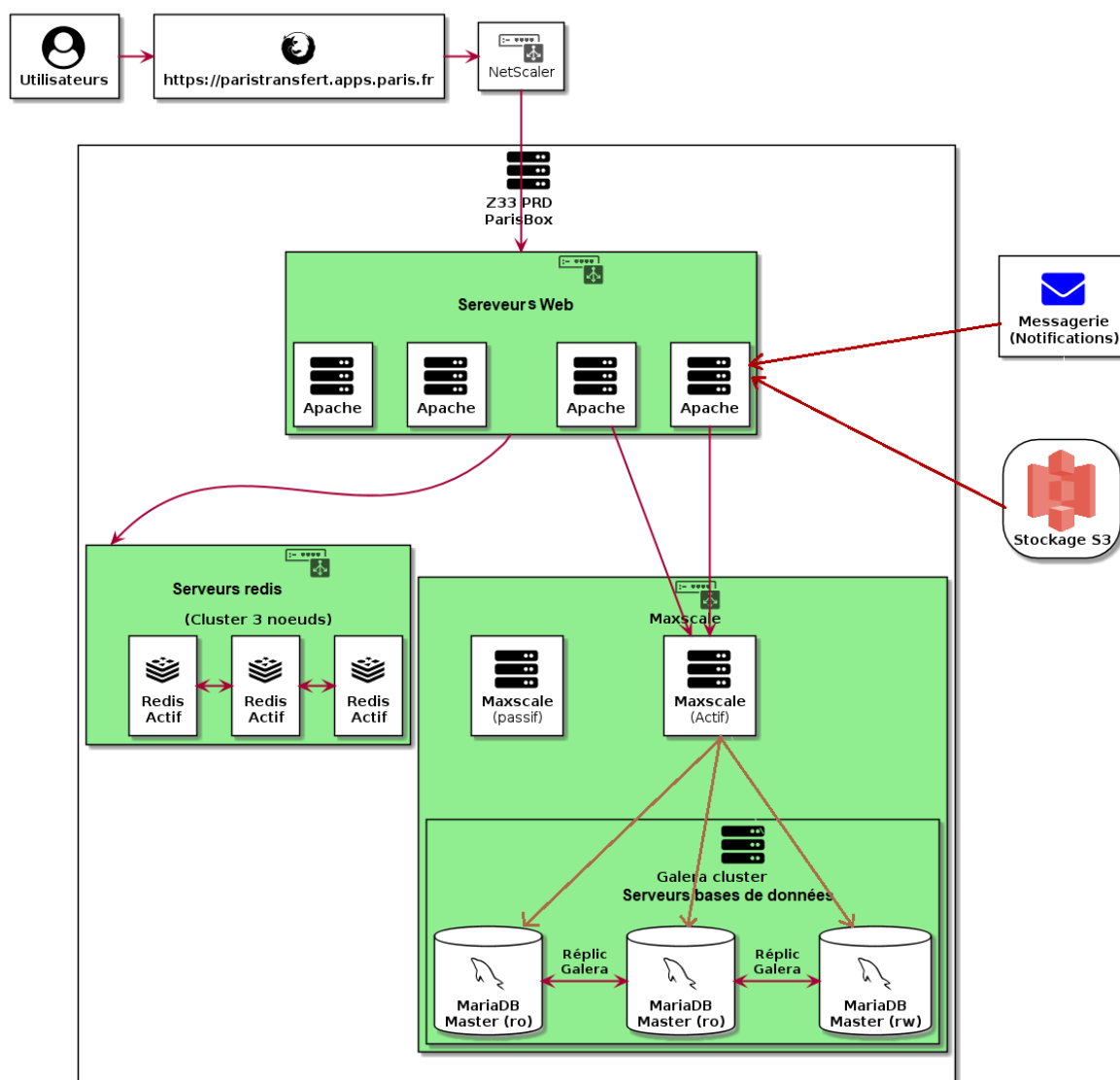


Figure 16 : nouvelle architecture NextCloud

Comme le montre la figure 15 l'architecture de l'application au niveau des serveurs web est la même que celle mise en place sur le 1^{er} POC, l'intérêt de mettre un cluster MariaDB Galera à la place d'un simple cluster PostgreSQL est la topologie du cluster qui est multi-maître actif-actif, elle permet de lire et écrire sur n'importe quel nœud de cluster et de contrôle automatique de l'appartenance, les nœuds défaillants sont supprimés du cluster.

Pour ce qui est du serveur dans lequel se trouve le montage NFS, nous avons privilégié un connecteur S3 pour sa rapidité et sa capacité qui sont beaucoup plus élevés qu'un montage NFS.

La nouvelle architecture de l'application sera appliquée par une entreprise de prestation tierce sur tous les environnements de développement à partir de mi-septembre et l'ouverture du service est prévue pour le mois d'avril 2021 comme le montre la figure 17 ci-après.

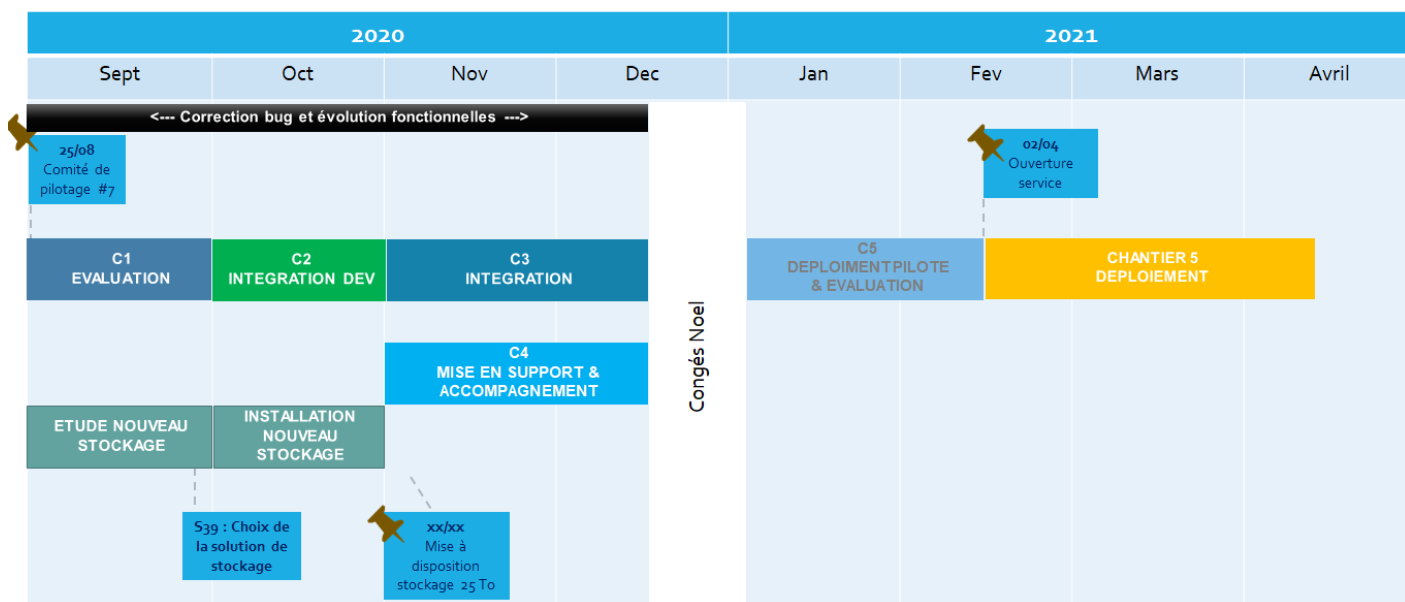


Figure 17 : planning de la mise en place de la nouvelle architecture

IX- Conclusion

L'objectif de ce travail effectué est de répondre aux besoins des agents de La ville de Paris et de pousser plus loin les limites des outils de communication de La Ville, en mettant en place une application qui permettra de trouver une solution définitive sur la problématique qui est l'impossibilité d'envoyer des fichiers volumineux avec les outils de communication actuelle.

Afin d'atteindre cet objectif nous avons travaillé en collaboration avec différents bureaux de la DSIN comme le Bureau des équipements et outils numériques « BEAON », Bureau de l'Exploitation, du Cloud et des Infrastructures Datacenter « BECID », etc. Cela m'a permis de développer encore davantage mon esprit de travail d'équipe et d'élargir mes connaissances dans d'autres domaines informatiques.

J'ai également pu entreprendre de nombreuses recherches individuelles sur l'architecture technique de l'application NextCloud, sur les différentes applications qui viennent renforcer notre application principale et les différentes contraintes à respecter, et dégager, de mon expérience en entreprise, un certain nombre de constats. Cela m'a permis d'émettre quelques réflexes sur la provenance des bugs et les méthodes à suivre pour corriger les erreurs.

De mon côté, j'ai pu aider l'équipe d'architecture dans l'installation, la configuration de toutes les applications qui composent le projet sur tous les environnements de développement, entre autres NextCloud, PostgreSQL, Redis, etc.

Le POC de l'application NextCloud a apporté ses fruits notamment sur la rapidité de réponses aux demandes d'utilisateurs, la possibilité de connexion en client lourd (SSO & SAML authentification), la synchronisation des contacts Outlook grâce à des plugins et la facilité de son utilisation néanmoins, il reste toujours des tests et des réunions à faire pour trancher sur la technologie comme l'utilisation du connecteur S3 pour le stockage des fichiers et le changement du cluster PostgreSQL en un cluster MariaDB.

Parmi les prochaines missions du BIAD, nous avons les missions qui suivent celle-ci dans l'ordre logique et chronologique tel que la publication définitive de l'application, une vérification et correction des éventuels bugs, le transfert de compétences à la Section Intégration des Services Numériques (SISN) pour garantir une bonne continuité du projet.

Ces deux années d'alternance m'ont permis de découvrir plus en profondeur le monde professionnel, les méthodologies de réalisation des projets, d'apprendre davantage sur le développement, l'intégration applicative, au métier d'Architecte technique et également de m'ouvrir à différents domaines comme le métier de chef de projet.

X- Bibliographie

- [1] Mitchell Anicas: *How To Install and Use PostgreSQL on CentOS 7* [en ligne]. Mise à jour le 08 janvier 2020. [Consulté en février 2020]. Disponible à l'adresse : <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-centos-7>
- [2] PlantUML : *PlantUML Documentation* [en ligne]. Mise à jour le 15 juillet 2019. [Consulté mai-août 2019]. Disponible à l'adresse : <http://plantuml.com/fr/>
- [3] NextCloud : *Admin manuel* [en ligne]. Mise à jour le 8 août 2020. [Consulté janvier-août 2020]. Disponible à l'adresse : https://docs.nextcloud.com/server/19/admin_manual/
- [4] NextCloud : *Application NextCloud* [en ligne]. Mise à jour le 8 août 2020. [Consulté mars 2020]. Disponible à l'adresse : <https://apps.nextcloud.com/>
- [5] NFS : *Montage de systèmes de fichiers NFS* [en ligne]. Mise à jour 1 novembre 2018. [Consulté juin 2020]. Disponible à l'adresse : https://doc.fedora-fr.org/wiki/Partage_de_disques_en_réseau_avec_NFS
- [6] WSSO : *Configuration WSSO* [en ligne]. Mise à jour le 14 août 2020. [Consulté en juillet 2020]. Disponible à l'adresse : https://help.highbond.com/helpdocs/highbond/fr/Content/launchpad/admin/configuring_sso_settings.html
- [7] Redis : *Documentation Redis* [en ligne]. Mise à jour le 2 août 2020. [Consulté en août 2020]. Disponible à l'adresse : <https://redis.io/documentation>
- [8] Pgwatch : *installation pgwatch* [en ligne]. Mise à jour le 13 août 2020. [Consulté en août 2020]. Disponible à l'adresse : <https://github.com/cybertec-postgresql/pgwatch2>
- [9] Influxdb : *installation Influxdb* [en ligne]. Mise à jour le 20 août 2020. [Consulté en août 2020]. Disponible à l'adresse : <https://www.influxdata.com/developers/>
- [10] Grafana : *installation Grafana* [en ligne]. Mise à jour le 15 août 2020. [Consulté en août 2020]. Disponible à l'adresse : <https://grafana.com/docs/grafana/latest/installation/rpm/>

XI- Table d'annexes

<i>Annexe A - Liste complète des plugins activé/désactivé</i>	<i>1</i>
<i>Annexe B - Architecture globale de la préproduction</i>	<i>2</i>
<i>Annexe C - MariaDB Galera Cluster</i>	<i>3</i>
Caractéristiques.....	3
Avantages	3

Annexe A - Liste complète des plugins activé/désactivé

Activé	Désactivé
Activity	Accessibility
Brute-force settings	Auditing / Logging
Collaborative tags	Comments
Contacts	Default encryption module
Deleted files	External storage support
File access control	Federation
File sharing	First run wizard
Files automated tagging	Nextcloud announcements
Impersonate	Photos
LDAP user and group backend	Recommendations
Log Reader	Support
Monitoring	Versions
Notifications	Video player
Password policy	clients Desktop
PDF viewer	clients Android
Privacy	clients IOS
Retention	/
Right click	/
Share by mail	/
SSO & SAML authentication	/
Terms of service	/
Text	/
Theming	/
Update notification	/
Usage survey	/

Annexe B - Architecture globale de la préproduction

Ce schéma d'architecture représente toutes les briques qui composent le POC actuel de l'application NextCloud ainsi que celles qui composent l'application de supervision Pgwatch.

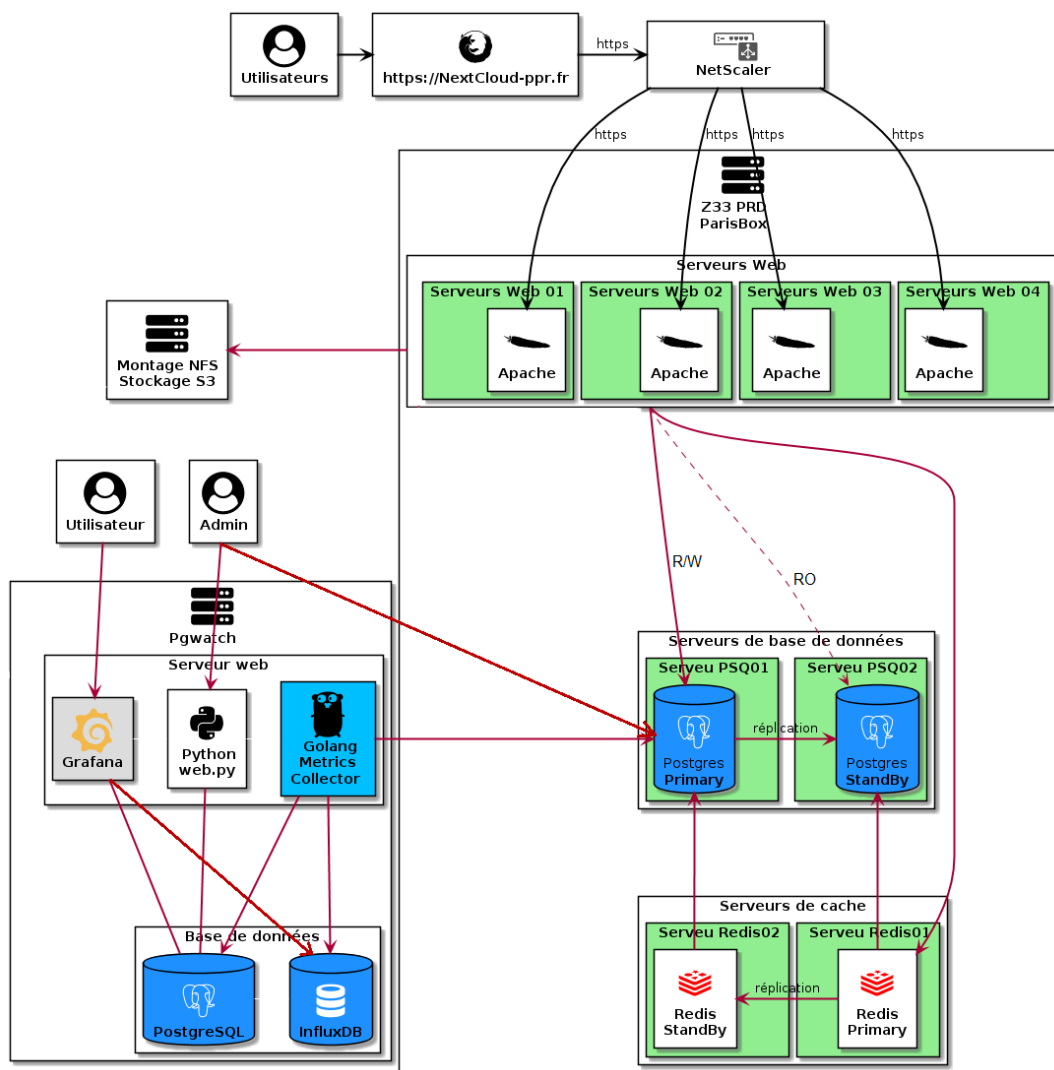
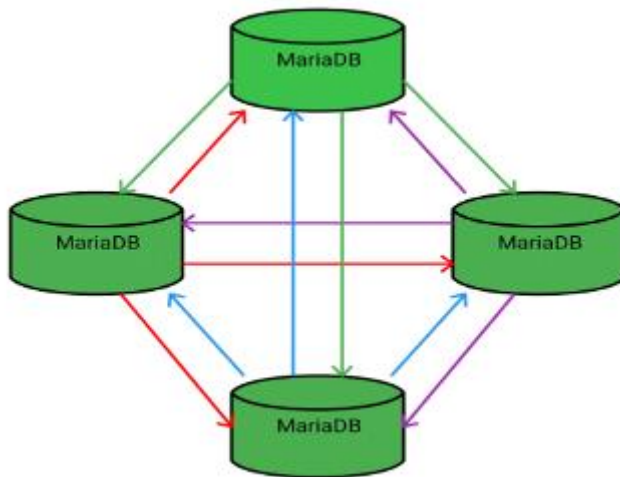


Figure 18 : Architecture global de la PPR

Annexe C - MariaDB Galera Cluster



MariaDB Galera Cluster est un cluster multi-maître pratiquement synchrone pour MariaDB.

Caractéristiques

- Réplication pratiquement synchrone
- Topologie multi-maître actif-actif
- Lire et écrire sur n'importe quel nœud de cluster
- Contrôle automatique de l'appartenance, les nœuds défaillants sont supprimés du cluster
- Jointure automatique des nœuds
- Véritable réplication parallèle, au niveau des lignes
- Connexions client directes, aspect et convivialité de MariaDB natifs

Avantages

Les fonctionnalités ci-dessus offrent plusieurs avantages pour une solution de clustering de SGBD, notamment:

- Pas de décalage esclave
- Aucune transaction perdue
- Lire l'évolutivité
- Moins de latences client