Dossier projet correction automatique

Cartographie_SI

Cahier des charges : Analyse du besoin

Introduction

La mairie de Val-de-Reuil, dans sa vision à long terme souhaite disposer d'une cartographie de son système d'information. Il n'existe actuellement pas en l'état un document, une méthode de ce genre permettant d'avoir une vue globale du système d'information.
L'état actuelle des lieux donne une vue suivante :
il existe un (1) Responsable de la modernisation numérique qui lui même travaille en étroite collaboration avec le Responsable de la Sécurité des Systèmes Informatique.
Ce dernier porte également la casquette d'administrateur des systèmes et du réseaux.
Il peut être assisté par un consultant extérieur pour ce qui est de la gestion des progiciels au compte de la mairie.

Une de mes tâches principale sera d'analyser et de comprendre le besoin, fournir une solution durable permettant à la ville de se munir d'un outil efficace, propriétaire et simple à utiliser.

Il existe un document élaborer par l'Agence Nationale de la Sécurité des Système d'information Guide ANSSI additionné à la Ebios risk manager qui donne une lecture de ce que devrait être une cartographie du SI.
Les Organismes d'importances Vitales (OIV) au sein d'un système d'information garantissent la continuité du SI. Leurs protections nécessitent une bonne lecture par les différents acteurs et responsables du système d'information. Une des raisons qui nécessite la mise en place d'une cartographie du SI est que lorsqu'il n'y a que quelques personnes qui connaissent le système d'information en soi cela peut être une faille de sécurité énorme pour l'organisme en question.

> Nul n'est à l'abri d'une erreur.

C'est pour cela que mettre en place une cartographie qui fera les connections nécessaires entre les différentes vues métiers, systèmes, et réseaux permettra de se munir d'outil efficace pour assurer une continuité des services, de par là favoriser une meilleur intégration des systèmes applicatifs au sein du système d'information.

En résumé, cette approche permettra à la ville et à ses responsables d'acquérir une maîtrise totale du système d'information. La mairie aura ainsi une connaissance exhaustive de tous les composants d'importance vitale du SI. Cette maîtrise se traduira par une meilleure

protection du système d'information, car nous serons en mesure d'identifier plus précisément les systèmes les plus critiques et les plus exposés. De plus, cela renforcera la capacité de défense de la ville, car une connaissance approfondie du SI permettra de réagir de manière plus efficace en cas d'incident ou d'attaque. Enfin, cette approche favorisera la résilience en permettant de mettre en place un plan de continuité d'activité, qu'il soit numérique ou non.

Plan de travail : analyse du besoin

Pour la réalisation de la cartographie j'ai mis en place une méthode de découpage de l'information au près des personnes ressources, cela a permis de dégager les étapes suivantes :

Collecte des informations :

- Réunir les documents existants sur l'infrastructure IT actuelle : Cette étape est cruciale et reste encore à parfaire, certaines données étant fortement sensible je n'ai pas eu accès.
- Rencontrer les différents responsables de service pour obtenir des informations supplémentaires.

Identification des composants du SI :

- Réseaux (LAN, WAN, VLAN).
- Serveurs (physiques, virtuels).
- Applications métiers (AxelNet, MobiliEcole).
- Bases de données (MySQL, PostgreSQL).
- Postes de travail (PC, tablettes).

Représentation graphique :
j'ai utilisé `Xmind` (outil de mindmapping) pour créer une représentation visuelle du SI décomposée en :

- Nœuds principaux : Réseaux LAN, WAN, VLAN, Serveurs (physiques, virtuels), Applications métiers (AxelNet, MobiliEcole), Bases de données (MySQL, PostgreSQL), Postes de travail (PC, tablettes).
- Sous-nœuds : détailler chaque composant en incluant des informations telles que les adresses IP, les versions logicielles, les interconnexions, etc.

Relations et interconnexions :

- Identifier et représenter les liens entre les différents composants du SI.
- L'utilisation des flèches de direction dans les tableaux donne une meilleur compréhension du flux de d'informations entre les données, les services et les composants du SI.

Documentation complémentaire :

- L'ajout des notes et des commentaires pour fournir des explications supplémentaires sur certains éléments du SI.
- j'ai inclu des informations sur la sécurité, la disponibilité et la performance de chaque composant selon les informations qui m'ont été fournies.

Mise à jour régulière :

- Une planification des sessions de mise à jour régulières pour maintenir la cartographie du SI à jour avec les évolutions et les changements. J'effectue pour cela un suivi sur les *issues github du code source*. S'assurer dans ce cas de bien lancer la commande de mise à jour de composer pour toutes les dépendances du projet avec la commande suivante : `composer update`

Partage et collaboration :

- Dans le plan de prise en main de l'outil de la cartographie, la gestion et les accès sont gérées par la définition des rôles :

1. L'administrateur : il a tout les droits. Il peut créer, modifier, attribuer de nouveaux rôle
2. Le Cartographe : il a des droits restreints sur les objets qu'il crée. Il n'a pas d'accès aux objets utilisateurs et configuration.
3. L'auditeur : a le droit d'accéder et d'afficher tous les objets exception faite des utilisateurs et de la configuration
   De nouveaux rôles peuvent être créer et définie selon les besoins de la municipalité.

Utilisation de la cartographie du SI :

- La carte est une aide à la prise de décision en identifiant les points faibles et les opportunités d'amélioration du SI.
- Elle se veut modératrice dans la communication et la compréhension entre les différents acteurs du SI.
- Elle doit servir de référence lors de l'élaboration de nouvelles stratégies ou de nouveaux projets.

Benchmark des communes ayant fait une cartographie de leur système d'information

je n'ai pas eu de référence dans mes recherches sur le net. Ce qui parait absolument censé car la cartographie représente l'ensemble du SI et donne une lecture globale pour qui comprend les aspects techniques. Ce qui m'a amené à me pencher d'avantage sur les différents outils disponibles pour créer une cartographie du SI.

Benchmark : Logiciels libres Architecture d'entreprise (AE)

N°	Nom	description	Avantages	Inconvenients	Notes
1	Modelio	modélisation de données	s'installe et se désinstalle facilement	Ne permet pas le partage entre plusieurs utilisateurs autorisés	testé
2	Visual Paradigm	Online	SAAS exclusif	Serveurs externes	testé
3	ADOIT	solution payante	clé en main	gestion externe et coût	testé
4	Arch Imatetool	excellent et opensource	gratuit puissant	courbe d'apprentissage	testé
5	Draw.io	hybride, locale et cloud	multiplateforme	Pas de prise en compte des recommendation ANSII	testé
6	Mercator	Norme ISO27001:2013 utilisé par 15 centres hospitaliers 3 communes en France.	née d'un besoin spécifique, en lien avec les récommandations ANSII	Suivi des Mises à jour, veille obligatoire	utilisé

Les vues

Les vues sont progressives allant du métier vers la technique, elles mêmes déclinées en vues. Ainsi tous les acteurs qui seront référencés peuvent avoir une lecture du SI. Les données peuvent être exportées sous plusieurs formats bureautique à des fins d'analyse. Formats disponibles : CSV, PDF, Excel.
Pour les besoins d'impression nous pouvons également définir les colonnes qui nous intéressent afin de les exploiter.

Métiers : Services de la Commune

!!! info "Ecosystème"

Vue d'ensemble : Processus => technologies utilisées => données produites => personnes (acteurs internes et ou externes)

Vue métier du SI : Processus et informations clés qui sont nécessaires pour que le SI remplisse sa fonction = Valeurs Métiers

Tableau des Métiers : Vue Métier

!!! infor "Vue métier et granularité"

niveau 1 : informations indispensable

niveau 2 : informations importantes

niveau 3 : informations utiles

Nom	Description	Responsable	@mail	granularité
Finances	Administration et Finances	Resp	@valdereuil.fr	1
Ressources Humaines	Administration et Finance	Resp	@valdereuil.fr	3
Centre Communal d'action social (CCAS)	Directions Opérationnelles	Resp	@valdereuil.fr	1
Enfance-Jeunesse-et-emploi	CCAS	Resp	@valdereuil.fr	1

Tableau services : Vue Processus

!!! info "Vue Processus"

j'ai commencé par décomposer la gestion de la paie :

identifier les Besoins et processus utilisés pour effectuer la paie.

Services	Processus	Logiciels
Gestion des ressources humaines	-Paie -Carrière -Agents -Retraite -Congés	x
Enfance et Jeunesse	Inscription -Revenus -Données sur la santé des enfants	x
Tiers, Subvention Budget	Données_Financières_et_Bancaires - Données_Personnelles -Données_Entreprises	x

Tableau : Vue Application

!!! info "Vue applicative"

Décrire les composants du logiciels du SI, des services offerts, et les flux de données entre eux.

La vue de l'administration : Nous allons répertorier les périmètres et les niveaux de privilèges c'est à dire les types d'utilisateurs(Cartographe, auditeurs, utilisateurs) et admin du SI.

Applications/Logiciels	Services A	Service B	Données
Horoquartz -> 2	Gestion des Ressources Humaines-> 3	Finances 4	<-temps/agents 1
AxelNet -> TeamNet	portail_famille->Enfance-jeunesse-Emploi hygiène	Finances	reservation, paiement, activités scolaires

Tableau : Vue Infrastructure

!!! info "Vue Infrastructure"
La vue des Infra Logiques : Cloisonnement logique des réseaux, la définition des plages d'adresses IP, les VLAN, WLAN,IPBX et les fonctions de filtrages utilisées sur les routeurs.
La vue des Infra Physique : Description des équipements physiques utilisés par le SI

Services	Plage @IP	Nom ou code VLAN	Equipements	Licences
Finances		1	PC, Mac, Workstation	O365
Ressources Humaines		1	PC, Workstation	
Centre Communal d'action social (CCAS)		1	PC, Workstation	
Enfance-Jeunesse-Petite Enfance CCAS		1	PC, Workstation	
Services Informatiques			Serveur physique : constructeur	

Application : Mercator pour la cartographie du SI

Mercator est un outil open source permettant de mettre en place une cartographie du système d'information. Il respecte le guide mis en place par l'ANSSI et a été développé par Didier Barzin. La dernière version est `Maturity 1c`. C'est l'outil qui répond le plus au besoin de la municipalité. L'historique du développement du projet est assez similaire à celui de la ville de Val-de-Reuil. Un besoin d'avoir l'information précise de suivi et d'évolution du SI par une petite équipe pour un grand nombre de matériel, logiciel, et projet en cours.
Dans le cas de la mairie j'ai choisi un serveur Fedora car il est robuste et incorpore les fonctionnalités avant-gardistes dans l'univers des serveurs linux. Il est dérivé de RedHat, sa documentation est à jour et bien concise.

Prérequis

Avant de commencer, il faut s'assurer de disposer des éléments suivants :

- Un serveur de production avec les caractéristiques suivantes :
- Système d'exploitation : Fedora Server 64 bits (un serveur dérivé de debian aussi peut parfaitement être mis en place)
- RAM : 2 Go

- Disque : 20 Go
- VCPU : 2

Installation

Installation et configuration d'un serveur fedora. La documentation officielle fournit une meilleur explication sur les différentes méthodes d'installation. Dans ce projet je travaille dans un environnement virtuelle sous Vmware workstation v16.
Ma configuration : 2Mo pour le Bios, 2 Go de swap, et 18 Go xfs sur la partition /.

Mis à jour : Fedora Linux (OS)

Avant d'installer Mercator, je recommande fortement d'effectuer une mise à jour votre système en exécutant la commande suivante :

```
dnf update --refresh -y
```

Installation des dépendances

Installez les dépendances nécessaires en exécutant les commandes suivantes :
Il s'agit pour l'essentiel de php, et d'un serveur web ici apache2.

```
  gd-2.3.3-10.fc38.x86_64                                graphite2-1.3.14-11.fc38.x86_64
  harfbuzz-7.1.0-1.fc38.x86_64                           highway-1.0.5-1.fc38.x86_64
  httpd-2.4.57-1.fc38.x86_64                             httpd-core-2.4.57-1.fc38.x86_64
  httpd-filesystem-2.4.57-1.fc38.noarch                  httpd-tools-2.4.57-1.fc38.x86_64
  jbigkit-libs-2.1-25.fc38.x86_64                        julietaula-montserrat-fonts-1:7.222-4.fc38.noarch
  jxl-pixbuf-loader-1:0.7.0-6.fc38.x86_64                langpacks-core-font-en-3.0-32.fc38.noarch
  libX11-1.8.6-1.fc38.x86_64                             libX11-common-1.8.6-1.fc38.noarch
  libXau-1.0.11-2.fc38.x86_64                            libXext-1.3.5-2.fc38.x86_64
  libXpm-3.5.15-3.fc38.x86_64                            libXrender-0.9.11-2.fc38.x86_64
  libaom-3.6.1-1.fc38.x86_64                             libavif-0.11.1-7.fc38.x86_64
  libdav1d-1.2.1-1.fc38.x86_64                           libimagequant-2.17.0-4.fc38.x86_64
  libjxl-1:0.7.0-6.fc38.x86_64                           libtiff-4.4.0-5.fc38.x86_64
  libvmaf-2.3.0-5.fc38.x86_64                            libwebp-1.3.1-1.fc38.x86_64
  libxcb-1.13.1-11.fc38.x86_64                           libzip-1.9.2-3.fc38.x86_64
  mod_http2-2.0.11-2.fc38.x86_64                         mod_lua-2.4.57-1.fc38.x86_64
  oniguruma-6.9.8-2.D20220919gitb041f6d.fc38.1.x86_64    php-common-8.2.9-2.fc38.x86_64
  php-gd-8.2.9-2.fc38.x86_64                             php-ldap-8.2.9-2.fc38.x86_64
  php-mbstring-8.2.9-2.fc38.x86_64                       php-mysqlnd-8.2.9-2.fc38.x86_64
  php-pdo-8.2.9-2.fc38.x86_64                            php-pecl-xdebug3-3.2.2-1.fc38.x86_64
  php-pecl-zip-1.21.1-2.fc38.x86_64                      php-soap-8.2.9-2.fc38.x86_64
  php-xml-8.2.9-2.fc38.x86_64                            rav1e-libs-0.6.6-1.fc38.x86_64
  svt-av1-libs-1.4.1-2.fc38.x86_64                       xml-common-0.6.3-60.fc38.noarch

Terminé !
[root@vdrMercator ~]# systemctl status httpd
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/service.d
             └─10-timeout-abort.conf
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@vdrMercator ~]# sys
```

```
dnf install php-zip php-curl php-mbstring php-dom php-ldap php-soap php-xdebug php-gd php-fpm php-mysqlnd httpd
```

Fedora : Service Apache

Démarrez le service Apache et configurez-le pour qu'il démarre automatiquement au démarrage du système :

```
systemctl start httpd

systemctl enable httpd
```

SGBDR : Installation de MariaDB

Installez MariaDB en exécutant la commande suivante :

```
dnf install mariadb-server
```

Démarrez le service MariaDB, vérifier et configurer pour qu'il démarre automatiquement au démarrage du système :

```
systemctl status mariadb

systemctl start mariadb

# Je vais rendre le service persistant grâce à la commande suivante :

systemctl enable mariadb.service
```

Installation de Git et Composer

J'installe Git, Graphviz et Composer; git pour le versionning, graphviz bibliothèque pour les vues et composer pour la gestion du site avec le framework Laravel.

```
Installé:
  adobe-mappings-cmap-20230622-1.fc38.noarch                 adobe-mappings-cmap-deprecated-20230622-1.fc38.noarch
  adobe-mappings-pdf-20190401-3.fc38.noarch                  cairo-gobject-1.17.8-4.fc38.x86_64
  composer-2.5.8-1.fc38.noarch                               cups-libs-1:2.4.6-4.fc38.x86_64
  fribidi-1.0.12-3.fc38.x86_64                               git-2.41.0-1.fc38.x86_64
  git-core-2.41.0-1.fc38.x86_64                              git-core-doc-2.41.0-1.fc38.noarch
  google-droid-sans-fonts-20200215-15.fc38.noarch            graphviz-7.1.0-3.fc38.x86_64
  gts-0.7.6-44.20121130.fc38.x86_64                          jbig2dec-libs-0.19-8.fc38.x86_64
  lasi-1.1.3-10.fc38.x86_64                                  lcms2-2.15-1.fc38.x86_64
  libICE-1.0.10-10.fc38.x86_64                               libSM-1.2.3-12.fc38.x86_64
  libXft-2.3.8-2.fc38.x86_64                                 libXt-1.2.1-4.fc38.x86_64
  libdatrie-0.2.13-5.fc38.x86_64                             libfontenc-1.1.6-2.fc38.x86_64
  libgs-10.01.2-3.fc38.x86_64                                libijs-0.35-17.fc38.x86_64
  libpaper-1:2.0.8-1.fc38.x86_64                             librsvg2-2.56.3-1.fc38.x86_64
  libthai-0.1.29-4.fc38.x86_64                               mkfontscale-1.2.2-3.fc38.x86_64
  netpbm-11.02.00-1.fc38.x86_64                              openjpeg2-2.5.0-3.fc38.x86_64
  pango-1.50.14-1.fc38.x86_64                                perl-Error-1:0.17029-11.fc38.noarch
  perl-File-Find-1.40-497.fc38.noarch                        perl-Git-2.41.0-1.fc38.noarch
  perl-TermReadKey-2.38-16.fc38.x86_64                       perl-lib-0.65-497.fc38.x86_64
  php-cli-8.2.9-2.fc38.x86_64                                php-intl-8.2.9-2.fc38.x86_64
  php-process-8.2.9-2.fc38.x86_64                            poppler-23.02.0-2.fc38.x86_64
  poppler-data-0.4.11-4.fc38.noarch                          poppler-glib-23.02.0-2.fc38.x86_64
  urw-base35-bookman-fonts-20200910-16.fc38.noarch           urw-base35-c059-fonts-20200910-16.fc38.noarch
  urw-base35-d050000l-fonts-20200910-16.fc38.noarch          urw-base35-fonts-20200910-16.fc38.noarch
  urw-base35-fonts-common-20200910-16.fc38.noarch            urw-base35-gothic-fonts-20200910-16.fc38.noarch
  urw-base35-nimbus-mono-ps-fonts-20200910-16.fc38.noarch    urw-base35-nimbus-roman-fonts-20200910-16.fc38.noarch
  urw-base35-nimbus-sans-fonts-20200910-16.fc38.noarch       urw-base35-p052-fonts-20200910-16.fc38.noarch
  urw-base35-standard-symbols-ps-fonts-20200910-16.fc38.noarch urw-base35-z003-fonts-20200910-16.fc38.noarch
  xorg-x11-fonts-ISO8859-1-100dpi-7.5-35.fc38.noarch

Terminé !
[root@vdrMercator ~]#
```

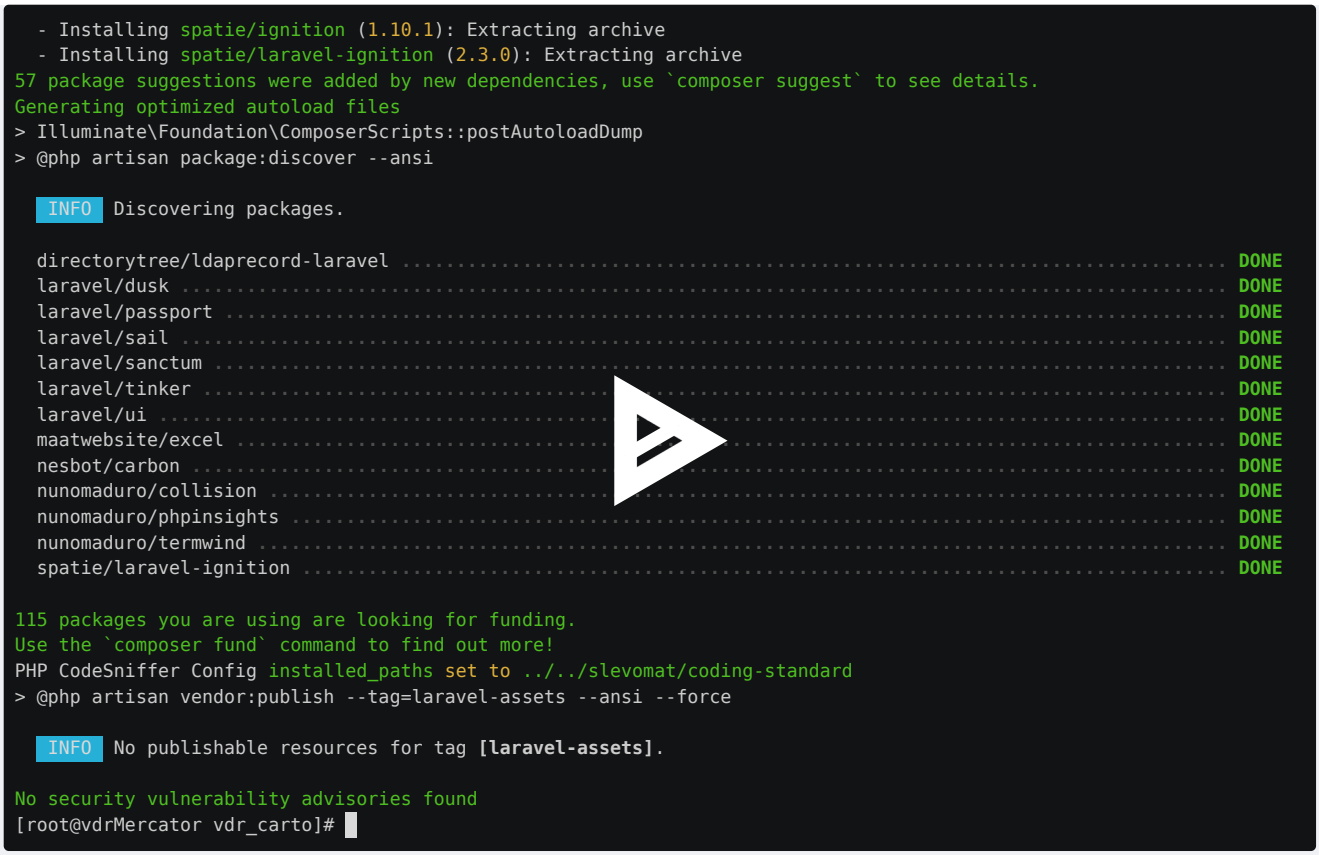
```
dnf install git graphviz composer -y
```

Mise en route du projet

Avant de cloner le projet sous github je crée un dossier pour le projet de la cartographie nommer par mes soins le plus simplement du monde et lui accorder des droits.

```
cd /var/www && mkdir vdr_carto && chown $USER:$GROUP vdr_carto
```

Ensuite, je clone le projet Mercator depuis GitHub en exécutant la commande suivante :




```
git clone https://www.github.com/dbarzin/mercator /var/www/vdrCarto
```

Accédez au répertoire du projet Mercator :

```
cd /var/www/vdCarto/mercator
```

Mettez à jour les dépendances de Composer en exécutant la commande suivante :

```
composer update
# au besoin
composer require
```

Publiez tous les paquets depuis le gestionnaire des paquets de PHP en exécutant la commande suivante :

```
php artisan vendor:publish --all
```

Configuration de la base de données

Création de la base de données

Il faudrait se connecter afin de pouvoir mettre en place une base de données MySQL.
En tant qu'utilisateur root et créez une base de données pour Mercator :

!!! warning "Sécurité production"

- Il faut créer un utilisateur, avec des droits administrateurs.
- Les commandes et la connexion ssh se feront avec ce compte et non en root
- Idem pour la gestion de la base de données.

```
mysql -u root -p
CREATE DATABASE mercator CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE USER 'vrd_user'@'localhost' IDENTIFIED BY 'xxxxxx';
GRANT ALL PRIVILEGES ON mercator.* TO 'vrd_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

```
MariaDB [(none)]> CREATE DATABASE mercator CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> CREATE USER 'vdr_admin'@'localhost' IDENTIFIED BY 's3cr3t';
Query OK, 0 rows affected (0,005 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON mercator.* TO 'vdr_admin'@'localhost';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> CREATE USER 'vdr_admin'@'10.110.12.158' IDENTIFIED BY 's3cr3t';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON mercator.* T      admin'@'10.110.12.158';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> exit;
Bye
[root@vdrMercator ~]# ls
anaconda-ks.cfg          creation-de-la-base-de-donnees  install-git-graphiz-composer
clone-mercator-from-git  installation-dependances-php    update-server
[root@vdrMercator ~]# cp /var/www/vdr_carto/.env.example .env
[root@vdrMercator ~]# vim /var/www/vdr_carto/.env
[root@vdrMercator ~]# vim /var/www/vdr_carto/.env.example
[root@vdrMercator ~]# vim /var/www/vdr_carto/ /var/www/vdr_carto .env
3 fichiers à éditer
[root@vdrMercator ~]# vim /var/www/vdr_cart
```

Configuration de Mercator

Créez un fichier `.env` à la racine du projet Mercator en copiant le fichier `.env.example` pour toutes les configurations ultérieurs nécessaire pour une connection à LDAP et SMTP : `cp .env.example .env`
Modifiez le fichier `.env` pour y ajouter les informations de connexion à la base de données.
J'utilise comme éditeur de texte vim : `sudo vim .env`

```
## .env file
DB_CONNECTION=mysql
# DB_CONNECTION=pgsql.env si nous utilisons postgresql
# DB_HOST=127.0.0.1 si nous restons sur notre machine en locale
DB_HOST=@ipduserveurvirtuel
# pour des raisons de sécurité, nous devons changer le port de la base de
données
DB_PORT=3306
# Comment DB_PORT for pgsql
DB_DATABASE=mercator
DB_USERNAME=vdr_admin
DB_PASSWORD=xxxxxx
```

Exécutez les migrations pour créer les tables de la base de données :

```
php artisan migrate --seed
```

Générez une clé d'application en exécutant la commande suivante :

```
php artisan key:generate
```

Nettoyez le cache des configurations en exécutant la commande suivante :

```
php artisan config:clear
```

Démarrage du serveur web

Démarrez le serveur web en exécutant la commande suivante :

```
php artisan serve -d
# Dans mon exemple en virtuelle pour accéder à l'hôte depuis mon addresse
il faut lancer la commande suivante
php artisan server --host 10.110.XX.XXX --port 8000
```

Vous pouvez maintenant accéder à l'application depuis votre navigateur à l'adresse http://localhost:8000. Sinon en virtuelle comme dans mon cas de figure sur l'adresse de votre hôte virtuel sur le port 8000.

Prise en main

Pour la première connection il faut utiliser les identifiants suivants :

!!! important "username et password"

- `admin@admin.com`

- `password`

Une fois connecté, vous pouvez commencer à remplir et administrer les données internes du système.
Pour des raisons évidentes de sécurité nous allons créer un nouvel admin, qui gérera les utilisateurs et la configuration du système. N'oubliez pas de bien supprimer celui de l'exemple.
Toutes les données seront rentrées manuellement, ce qui implique que chaque donnée collectée est bien maîtrisé avant son intégration dans la cartographie.

Sécurité: Serveur et de la base de données

1. Serveur :
   Lister tous les services en cours d'exécution afin de déterminer les services qui ne sont pas utiles pour l'exécution de notre base de données afin de limiter les surfaces d'attaques.

```
sudo dnf lsof -i
# lister les processus en cours d'exécution
sudo systemctl | grep running
# analyser et arrêter les processus qui ne sont pas utiles à l'éxécution du système
sudo systemctl stop bluetooth.service && sudo systemctl disable bluetooth.service
```

2. Connection ssh
   Editer le fichier de configuration, changer le port d'accès ssh. Interdire la connection ssh via superuser (root). Il faut créer un utilisateur spécifique ou un group spécifique pour la connection au serveur.
   Restreindre les accès à une plage d'adresse spécifique : `192.168.x.x dans le fichier etc/ssh/sshd_config`
3. Préparer un cron pour la sauvegarde :
   J'ai mis en place un petit script pour les sauvegardes de la base de données.

!!! Important "mysql-dump priviliges"
En lançant le script il peut y avoir un message d'erreur de connexion à la base de données pour réaliser le backup
Solutions : reload & process

```
GRANT RELOAD, PROCESS ON *.* TO 'vdr_admin'@'%';
# reload privilèges
FLUSH PRIVILEGES;
# pour voir la liset des utilisateurs et des hôtes
select `User`, `HOST`, `Process_priv`, `Reload_priv` FROM mysql.user;
```

```
#!/bin/bash
# Configuration de la date pour le dossier de sauvegarde
NOW=$(date +%Y%m%d%H%M)
echo $NOW
backupDir="/home/fmalo/plan_de_sauvegarde/$NOW"
# Création d'un nouveau dossier avec la date
mkdir $backupDir
# Sauvegarde de la base de données MariaDB
mariadb-dump -u vdr_admin -pnu11pvdr -B mercator -x -e --lock-tables --
flush-logs > "$backupDir/vdrcarto.sql" | gzip -9
"$backupDir/vdrcarto.sql.gz"
```

`crontab : 0 0 * * 1,5`

activer le cron avec la commande `crontab -e`

!!! info "Explication du script"
Ce script bash effectue les opérations suivantes :
Il génère une date au format "YYYYMMDDHHMM" et la stocke dans la variable NOW.
Il crée un nouveau répertoire de sauvegarde avec le nom de la date générée.
Il utilise la commande mariadb-dump pour sauvegarder la base de données "mercator" dans un fichier "vdrcarto.sql" situé dans le répertoire de sauvegarde.
Il compresse le fichier "vdrcarto.sql" avec gzip, en obtenant "vdrcarto.sql.gz" dans le même répertoire de sauvegarde.
Pour automatiser cette sauvegarde, j'ai configuré une tâche cron qui exécute ce script à 15 heures, les lundis et vendredis de chaque semaine.

Modélisation: Système d'information

J'ai fait une modélisation préalable sous yED, sous la machine qui m'a été donné durant le stage. yEd est un logiciel puissant qui permet de la modélisation du système d'information. Il n'en demeure pas moins qu'un outil comme mercartor répond mieux à toutes les préoccupations d'un responsable de la cartographie du SI.Mercator est bien plus qu'un outil d'inventaire il permet de comprendre les relations métiers et informatique de façon plus approfondie.

Suivi et mise à jour du système :

git pull : mise à jour en locale depuis le dépôt distant

Dans le dossier var -> www -> vdr_carto lancer la commande `git pull` && `composer update`
cette tâche peut être automatiser dans un script. Dans le même état d'esprit nous pouvons affiner les rôles, cette tâche reviendra à l'administrateur.

Comprendre la cartographie : IMPORTANT

Règlement Général sur la Protection des Données (RGPD)

1. Registre : Dans cette section, nous explorons le registre RGPD en détail. Nous indiquons le nom du registre, sa description, le responsable du traitement, la finalité du traitement, les destinataires, la durée de rétention, le processus impliqué, les applications associées, et le cas échéant, nous chargeons un document référentiel
2. Mésure de sécurité : Nous allons examiner ici les méthodes de suivis ou de vérifications de la conformité au RGPD qui ont été mises en place par les services responsables. L'implication du Délégué à la Protection des Données est capitale.

Système d'information

1. Macro-Processus : Cette section vise à définir les macro-processus avec des noms explicites, en décrivant les éléments entrants et sortants. Indiquer les besoins de sécurité en tenant compte des indicateur suivants : le niveau de confidentialité, son niveau d'intégrité, son niveau de Disponibilité, et enfin le niveau de Traçabilité (`CIDT`). Il faut également définir un propriétaire qui peut être une entité/services, et lui attribué les processus découlant de la macro.
2. Processus : décrit l'activité soutenu par une ou plusieurs entités/Services. Il est important de situé les informations qui seront utiles dans le système d'information d'un point de vue granularité de l'information. il faut indiqué le ou les types d'activités, les applications soutenues c'est à dire celles qui soutiennent le processus. Identifier le type d'information traité par le processus ex: données bancaires, données personnelles. Nommer le responsable, et tagguer la macro. Définir les besoins de sécurité du processus.
3. Activités : Il s'agit de nommer l'activité de traitement, faire une description du traitement. Identifier le ou les processus lié(s). S'il existe des opérations en cours qui sont liés, nous les identifions.
4. Informations : Dans la cartographie l'information doit être identifiable, liée à un processus, avoir un propriétaire, un administrateur. Nous devons également indiquer son format de stockage, sa sensibilité, et ses besoins de sécurités du point de vue CIDT. S'il existe une contrainte réglementaire il faut le préciser et indiquer la norme en question

Ecosystème

1. Entités : Indispensable pour la compréhension de la vue métier dans la cartographie, l'entité c'est ou le service qui participent à l'activité de traitement, manipulation des processus métiers en étroite lien avec les applications métiers identifiées. L'entité peut être interne ou externe. Nous l'identifions par son nom administratif, son niveau de sécurité, c'est à dire le type de maturité, le point de contact, le lieu de stockage des données manipulées par l'entité.

2. Relations : En définissant la nature de la relation qui existe entre les entités, que ce soit des biens, des servies, un partenariat commercial, nous avons une vue claire des différentes interactions qui donnent une lecture sur l'impact que peut avoir une application dans sa globalité.

Applications

1. Bloc Applicatif : Il s'agit d'identifier le fournisseur de l'application.
2. Applications : Les applications métiers utilisées. il faut décrire l'écosystème de l'application : le responsable, le ou les entités qui utilisent l'application, nommé un référent fonctionnel, déterminer l'éditeur, le volume d'utilisateur de l'application, le cartographe (la personne qui renseigne l'application). Nous allons également exposer l'aspect technique de l'app. Définir le type de client c'est à dire cloud, web, on-premise, si l'app est un progiciel, en développement interne, son exposition à l'externe type de solution SAAS, ou autre. Il faut aussi pour un suivi et mis à jour de l'app indiquer la date d'installation, la date de la dernière mise à jour, la documentation technique. Identifier et indiquer la liste des services applicatifs délivrés par l'application, la liste des bases de données utilisées par l'application. Son niveau de sécurité d'un point de vue CIDT.Très important c'est la possibilité de définir un plan de continuité et/ou de reprise d'activité Recovery time objective (RTO) et le Restart time objective. Du point de vue sécurité de l'app nous avons la possibilité d'uploader une base de données de CVE sur la plateforme.Voir les commandes suivantes pour l'intégration d'une base disponible pour le common plateforme Enumeration CPE `gzip -d mercator_cpe.sql.gz` && `sudo mysql mercator < mercator_cpe.sql`. Faire le lien avec le système d'information et l'infrastructure logique qui héberge la solution.

Infrastructure logique

0. Entités : Point d'entrée globale
1. Réseaux : Nom du réseau identifié qui comportera une description, le type de protocole, le responsable de l'exploitation, le responsable de sécurité du système d'information, les besoins de sécurités (CIDT)
2. Sous-réseaux : Un nom pour identifier notre premier sous-réseaux, sa description, la plage d'adresse et le masque de sous réseaux, sa zone de pare-feu, la passerelle par défaut, son appartenance à un Vlan, le type d'allocation (dynamique ou statique)+ la passerelle (NAT) son réseau de référence, préciser s'il accède au Wi-fi ou non. Si nous avons mis en place une zone démilitarisé nous allons également le spécifié.
   Une vue concise de l'infra logique dépendra de la description du réseau de l'entreprise, si elle n'est pas claire la gestion du risque de sécurité peut s'avérer délicate.
3. Serveurs Logiques : Établir la liste des serveurs logiques : Linux, Windows Server (année), hyperviseurs et Virtual Machines
4. Routeurs : Il faut spécifier le type de routeur, son nom, ses caractéristiques techniques et son adresse IP. Nous allons également préciser les règles de filtrages appliqués sur

le routeur (gestion des Vlans)

5. Commutateurs : le nom de notre switch , sa description, son adresse IP, et ses caractéristiques techniques
6. Vlan : Il existera autant de vlan que nécessaire pour une meilleur gestion qui doit être représenter dans la cartographie. il faut indiquer l'ensemble des sous-réseaux qui le compose
7. Certificats : la gestion des certificats permet d'envoyer à la personne responsable des certificats d'être prévenue dans les délais prévues pour prendre les mesures nécessaires.

Infrastructure Physique

1. Sites : Peut être le découpage géographie ou un découpage administratif, dans l'exemple de la municipalité il n'existe qu'un site physique c'est la ville. J'ai pris sur moi suite aux explications que j'ai reçu de part le RSSI de découper en deux sites : A -> B. A pour la gestion des Bâtiments connectés qui se situent sur pluisieurs zones. B normalement devrait regroupé les bâtiments non encore connecté de la mairie. Ainsi si un bâtiment doit être connecté à l'avenir son plan d'intégration au réseau existant peut être facilement réalisé.
2. Bâtiments/Salles : L'ensemble des 8 bâtiments connectés identifiés dans la ville.
3. Postes de travail, Périphériques, Borne Wifi, Téléphones : il s'agit ici de faire l'inventaire de l'ensemble des équipements informatiques de la ville.
4. Baies : Serveurs physique, Commutateurs, stockage, Routeurs, Sécurité : idem réaliser l'inventaire exhaustif des acquisitions physique du matériel informatique.

Workflow : amelioration continue

- [x] ~~Présentation PPTX~~
- [x] ~~Documentation en ligne~~
- [x] ~~Suivi et mis à jour du système~~
- [ ] Choisir les futurs cartographers
- [ ] Définir les auditeurs du système
- [ ] Formation et présentation au différents services (sensibilisation)
- [x] ~~Veille Common vulnerabilities and exposures cve-search~~
- [ ] Backup vm et données de la base
- [ ] Script pour la conservation et la suppression automatique des sauvegardes.
- [ ] Config : web server(Nginx)/ou Apache
- [ ] Générer un nouveau certificat, si non utilisé un certificat disponible: Transport Layer Secure/Secure Socket Layer connection (TLS/SSL).