

Ruiz Guillaume

STS SN1

RAPPORT DE STAGE

Entreprise : Odéolis

14 mai au 22 juin 2018



Lycée Vauvenargues → 60 Boulevard Carnot, 13090 Aix-en-Provence
Lieu de stage → 230 Rue Frédéric Joliot, 13100 Aix-en-Provence

Nom du tuteur : Laurent Friedrich

SOMMAIRE

Introduction.....	2
Remerciements	2
L'entreprise.....	3
Présentation	3
Service d'accueil.....	7
Activités.....	9
Contexte général.....	9
Cahier des charges.....	12
Préparation	13
Création du script.....	16
Limites et problèmes rencontrés.....	20
Conclusion.....	21
Annexes	22

INTRODUCTION

Dans le cadre de ma première année de section de technicien supérieur en système numérique j'ai réalisé un stage en entreprise de six semaines, du 14 mai au 22 juin 2018.

Ce stage s'est déroulé aux Milles (Aix-en-Provence), dans une entreprise du nom d'Odéolis offrant ses services à de grandes entreprises en gérant l'installation et la maintenance de parcs informatiques et d'équipements de libre service.

Ce stage m'a permis de découvrir les coulisses d'une entreprise travaillant dans le domaine de l'informatique. De plus j'ai pu y approfondir mes connaissances sur les architectures réseaux ainsi que les scripts de commandes systèmes.

REMERCIEMENTS

Chaque membre du personnel auquel je me suis adressé s'est toujours comporté de manière courtoise et chaleureuse, me permettant de passer un stage dans d'excellentes conditions, alors je tenais à leurs adresser mes plus sincères remerciements, à eux, ainsi qu'à toutes les personnes qui ont rendu ce stage possible ainsi que celles qui m'ont accompagné tout au long de celui-ci.

Dans cette optique, je souhaiterais dans un premier temps remercier Laurence Bugnone et Catherine Imbault, qui ont transmis et traité ma demande de stage, et m'ont chaleureusement accueilli au sein de l'entreprise.

Je souhaiterais aussi remercier les membres du service m'ayant pris en charge : Laurent Pizy, Laurent Friedrich (mon tuteur de stage), Jérôme Durand et Mourad Benbouchta, qui en plus d'avoir été d'une très grande sympathie avec moi, ont été très attentifs à mes besoins, à l'écoute, et toujours présents pour m'aider ou me documenter, souvent sans même avoir à leurs demander quoi que ce soit.

Pour finir, j'aimerais tout particulièrement adresser mes remerciements à Clément Sitouze, m'ayant pris en charge, et ayant sacrifié énormément de son temps afin de me former. Ce fut un véritable tuteur, partageant son expérience et son savoir, tout en restant une personne toujours extrêmement agréable.

L'ENTREPRISE

PRESENTATION

PRESENTATION GENERALE

Odéolis est une entreprise française indépendante. Son cœur de métier est d'assurer le fonctionnement sans interruption des équipements de transaction et de libre-service d'autres entreprises tels des distributeurs automatiques, des bornes de recharge pour voiture électriques, d'enregistrement Air France ou d'achat de jeux à gratter.



Odéolis se charge des :

- Remises en service
- Maintien en conditions opérationnelles
- Mises en service et déploiements
- Accompagnements d'évènementiels



10 millions
d'utilisateurs



25 ans
d'expérience



75.000
points de vente maintenus



350
collaborateurs



300.000
systèmes supportés



40M€
de chiffre d'affaires

Site internet : « www.odeolis.fr »

STRUCTURE JURIDIQUE

Renseignements juridiques :

→ Dénomination :	ODEOLIS
→ Adresse du siège social	101 rue Saint Lazare, 75009 Paris
→ Forme juridique INSEE :	Société par actions simplifiée
→ Activité (Code NAF ou APE) :	Activités informatiques (6209Z)
→ Date création entreprise :	12 mars 2010
→ Effectifs :	315 salariés
→ Capital social :	500 000,00 €
→ Chiffre d'affaires 2016	32 382 500,00 €

Dirigeants :

→ Président	Pierre-Yves BESEME
→ Directeur général	Grégoire de La Horie

HISTORIQUE

L'entreprise a été créée par en 1991, elle était alors une filiale annexe de La Française des Jeux, et s'appelait La Française de Maintenance. Comme son nom l'indiquait, l'entreprise gérant l'installation et la maintenance des équipements informatiques de la Française des Jeux.

Elle fut vendue en 1998 à une entreprise du groupe américain EDS et a été renommée EDS GFS. Puis vendue à nouveau à un groupe allemand, elle est devenue AO GROUP.

En 2010, elle fut renommé Odéolis après son rachat par des actionnaires dont Grégoire de La Horie, aujourd'hui devenu Directeur général de l'entreprise.

SITUATION GEOGRAPHIQUE

Odéolis est une entreprise Française présente sur l'intégralité du territoire national. Elle est composée de locaux techniques, dépôts et agences régionales, ainsi qu'une Plate-forme logistique (à Vitrolles) et un centre opérationnel (à Aix en Provence où j'ai effectué mon stage), disséminés sur toute la France comme illustré ci-dessous.

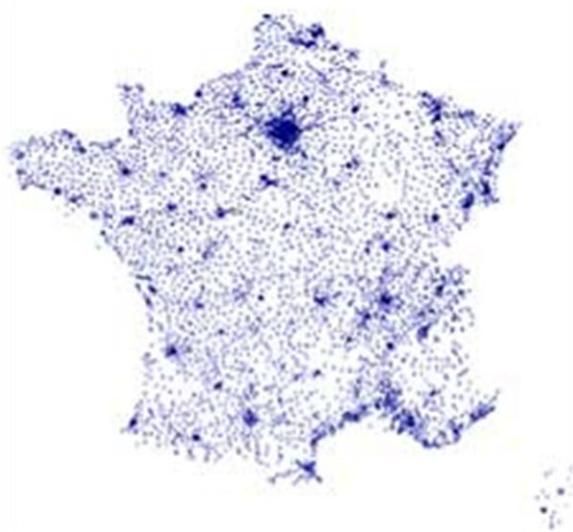


Odéolis en France

- 57 locaux techniques
- 12 dépôts régionaux
- 5 agences régionales
- Plate-forme nationale logistique, atelier & services supports

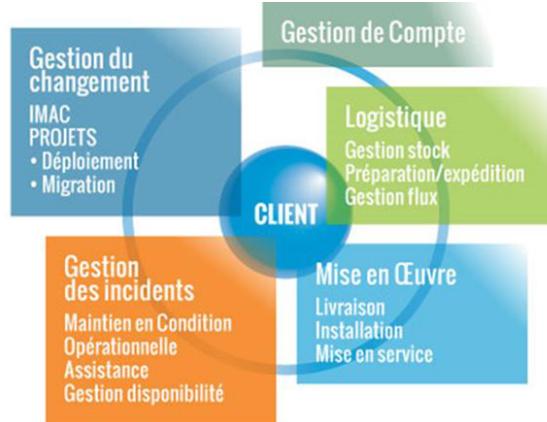
Grâce à l'historique de l'entreprise avec la Française des Jeux (le plus important réseau de points de vente français), leur maillage territorial ne cesse de croître (plus de 75 000 points de vente couverts).

Les moyens logistiques d'Odéolis permettent aux techniciens d'avoir les pièces de rechange de manière immédiate et jusqu'à moins de 3h00 des lieux d'intervention.



ACTIVITES ET CLIENTS

Services client :



Odéolis possède un large panel d'activités et de services (de gestion, logistique et mise en œuvre) pour un panel de clients tout aussi large, dont voici quelques exemples :



- | | |
|---------------------------|--|
| → Française des Jeux : | Terminaux et bornes |
| → PMU : | Terminaux, bornes et imprimantes |
| → SITA-AIR France : | Bornes d'enregistrement Air France |
| → DIEBOLD : | Automates bancaires |
| → Bimedia : | Terminaux de caisse pour buraliste |
| → Pari Mutuel Computing : | Terminaux de paris mutuels |
| → Morpho : | Terminaux de jeux et de paris |
| → LOGISTA France : | Terminaux, caisses, PC et imprimantes |
| → DBT-CEV : | Bornes de rechargement électriques |
| → INPOST : | Distributeur de colis en libre-service |

SERVICE D'ACCUEIL

PRESENTATION DU SERVICE

Le service auquel j'ai été assigné se nomme le « Service Desk », dirigé par Laurent Pizy. Le Service Desk est le service en relation avec la gestion des appels de clients, souvent des membres du personnel d'entreprises clientes.

Le type d'appel que j'ai le plus remarqué lorsque je travaillais dans l'open space, concernait des pannes de caisses de supermarché (service MCO → Maintien en condition opérationnelle).

Le service se segmente en trois parties, appelées niveau 1, 2 et 3.

Le niveau 1 reçoit les appels cités précédemment, ainsi que ceux de techniciens en intervention qui feraient face à un problème ou qui auraient besoins d'informations complémentaires sur les manipulations à effectuer (on appelle ces appels de techniciens des « escalades »).

Dans le cas où le problème rencontré s'avère plus profond et technique, l'appel est redirigé vers le « support technique » (niveau 2 ou 3).

A noter que le niveau 3 (dont fait partie Clément Sitouze, la personne s'étant occupé de notre activité de stage pendant ces six semaines) est composé d'experts ayant chacun leurs propres domaines d'expertises, qui en plus de travailler auprès du niveau 1 et 2, permettent la formation de jeunes techniciens.

PLACE DU SERVICE DANS L'ENTREPRISE

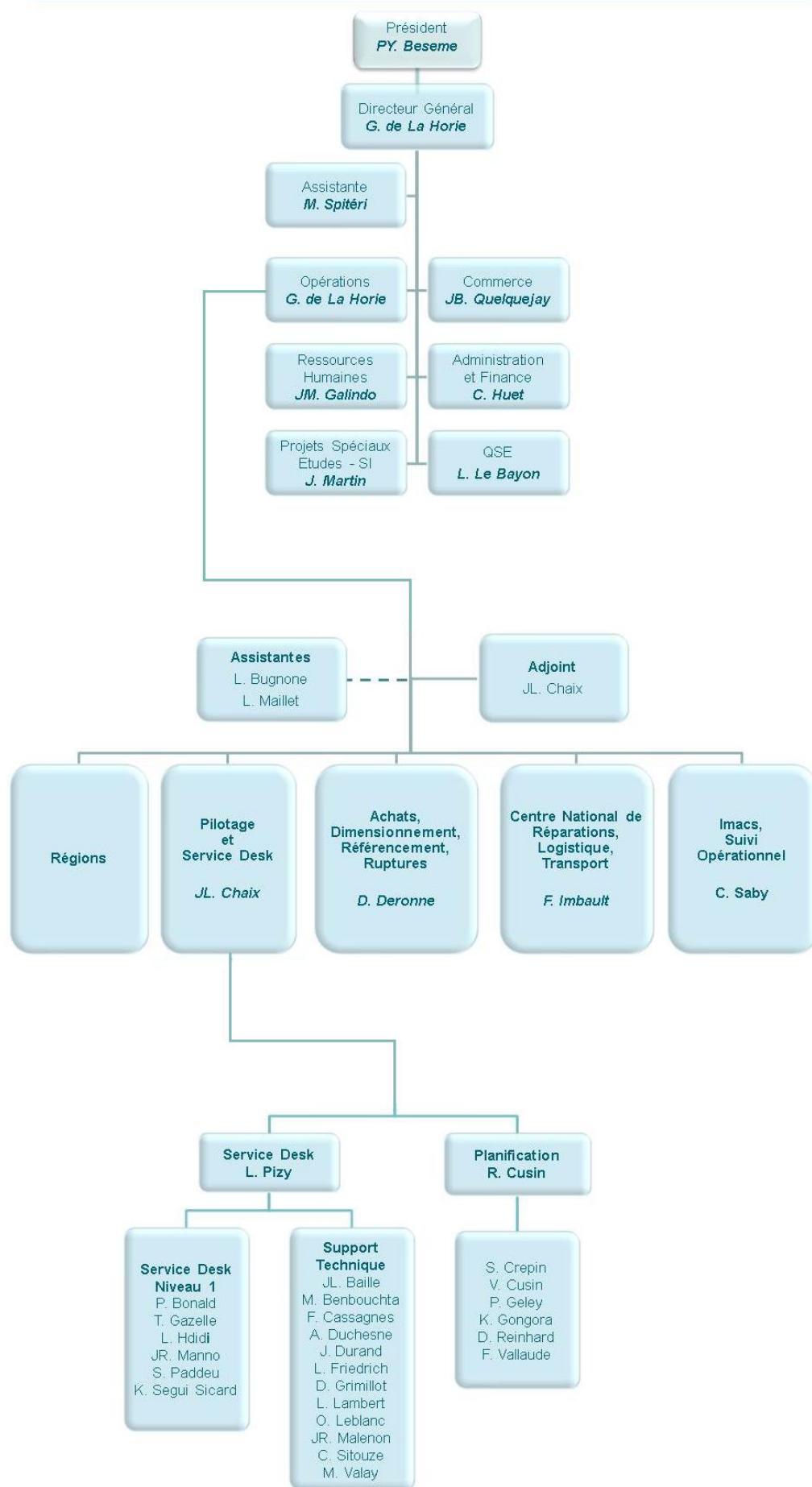
Le service desk n'est qu'un service parmi beaucoup d'autres au sein d'Odéolis.

Il fait partie du secteur des opérations, qui coexiste avec d'autre secteurs comme ceux dédiés au commerce, aux ressources humaines, à l'administration et aux finances etc...

Et au sein même de ce secteur des opérations, le service desk coexiste avec d'autres services tels que ceux gérant les installations, mouvements, ajouts et changements (service IMAC), l'achat, le dimensionnement et le référencement des pièces, la logistique, le transport, le suivi opérationnel etc...

Organigramme de la place du service desk au sein d'Odéolis :

Odéolis



ACTIVITES

CONTEXTE GENERAL

PRESENTATION DE LA STIME

Ma mission de stage était une reproduction d'une intervention technique effectuée il y a quatre ans pour client : la Stime, l'entité qui gère le coté informatique du groupe Les Mousquetaires.

L'enseigne Les Mousquetaires compte 7 filiales :

- Intermarché
- Netto
- Bricomarché
- Brico Cash
- Bricorama
- Roady
- Poivre Rouge



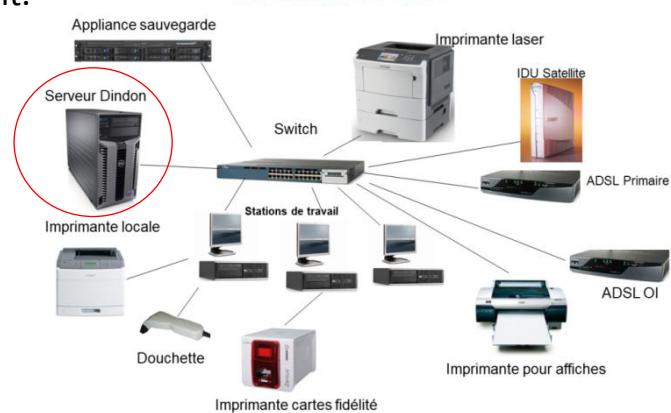
Tous les établissements de cette enseigne possèdent des parcs informatiques similaires dont l'installation et la maintenance sont gérées par Odéolis.

Ces parcs informatiques sont composés de deux parties, une partie « Caisses » (le Front Office) et une partie « Bureaux » (le Back Office).

ARCHITECTURE DINDON

L'opération effectuée par Odéolis il y a quatre ans concernait la partie « Bureaux » puisqu'il s'agissait d'un remplacement du serveur principal du Back Office par un modèle plus récent. Ce serveur, appelé « Serveur Dindon » permet la centralisation des applications et de la majeure partie des données de l'établissement.

Le Back Office



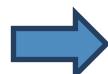
MISSION DE STAGE

Le remplacement de ce serveur nécessite de faire migrer les machines virtuelles émulées de l'ancien au nouveau modèle.

Pour ce faire, un script PowerShell permettant d'automatiser le processus de migration des machines virtuelles de l'ancien au nouveau serveur Dindon a été créé, de manière à être réutilisé quel que soit l'établissement du groupe Les Mousquetaires (projet Dindon).

Pour ce stage, ma coéquipière et moi avons travaillé en binôme. La mission de stage qui nous a été assignée consistait donc à développer un script PowerShell ayant la même fonction que celui créé il y a quatre ans.

Salle dans laquelle nous avons pu travailler. Elle contient plusieurs serveurs, switches, câbles et autres équipements dont s'occupe Odéolis. Cette salle s'apparentant à une salle de test, nous avons pu librement nous servir du matériel souhaité, et effectuer nos configurations sans risque particulier d'abîmer le matériel.



Les Serveurs Dindon

Dell T610



Dell T620



Dell T630



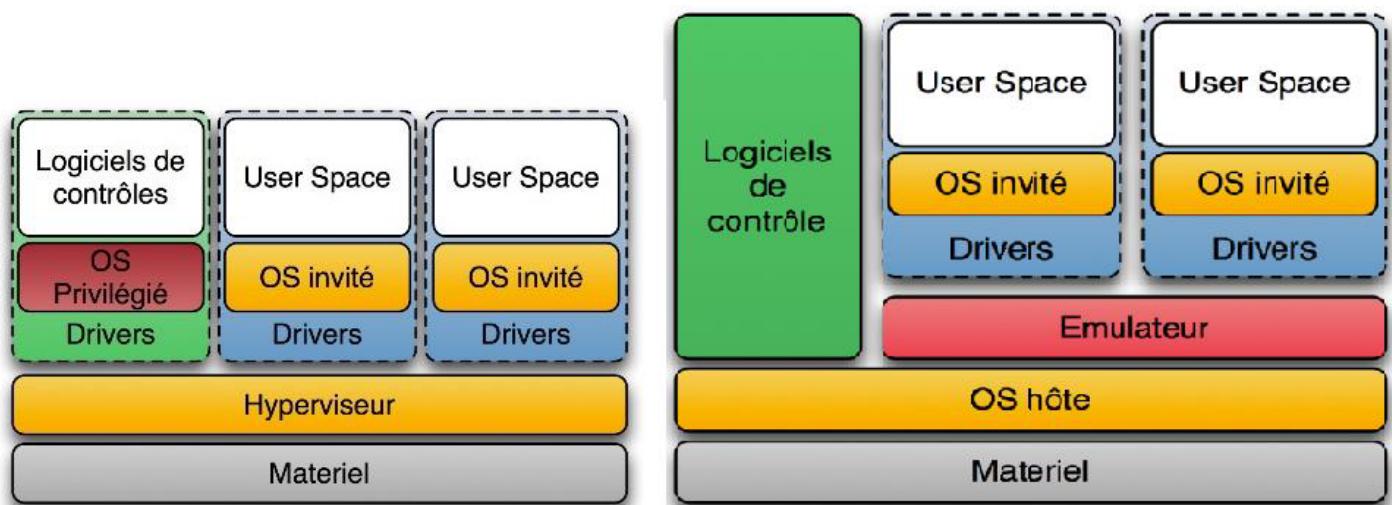
Dans notre cas, le serveur Dindon était un Dell T610 devenant obsolète et étant destiné à être remplacé par un Dell T630.

Sur ce serveur est utilisé l'hyperviseur Microsoft Hyper-V serveur 2012 (aussi appelé Windows Server Virtualisation) afin de virtualiser trois machines virtuelles.

Un hyperviseur est un outil de virtualisation qui permet à plusieurs systèmes d'exploitation de fonctionner simultanément sur une même machine physique.

Hyper-V est un hyperviseur de type 1. Contrairement à un hyperviseur de type 2, qui se présente comme un logiciel permettant d'émuler des machines virtuelles, le tout sur un système d'exploitation principal, un hyperviseur de type 1 est directement installé sur le matériel, gérant donc le système d'exploitation principal au même niveau que les machines virtuelles.

Cette caractéristique lui permet d'être très léger, puisque toutes les ressources matérielles sont directement utilisables pour les machines virtuelles, sans système d'exploitation intermédiaire.



Hyperviseur de type 1

Hyperviseur de type 2

CAHIER DES CHARGES

- IP réseau = 10.%octet2%.%octet3%.0 /24

→ Serveur Dell T610 :

→ Windows Serveur 2012 R2

→ Nom de machine : HV%codehex%01

→ Adresse IP : 10.%octet2%.%octet3%.36

→ Domaine : PDV%codehex%.mousquetaires.local

→ Hyper-V afin d'émuler les trois machines virtuelles suivantes :

	IF%codehex%01VM	SR%codehex%01VM	SR%codehex%02VM
Nom de machine	IF%codehex%01	SR%codehex%01	SR%codehex%02
Adresse IP	10.%octet2%.%octet3%.22	10.%octet2%.%octet3%.21	10.%octet2%.%octet3%.5
Rôles et services	AD / DNS / DHCP /WDS	X	X
Domaine	PDV%codehex%.mousquetaires.local	PDV%codehex%.mousquetaires.local	PDV%codehex%.mousquetaires.local

→ Nom des cartes réseaux :

- « NIC-01-P-CAISSE »
- « NIC-02-P-INFRA »
- « NIC-03-P-INFRAIGMP »
- « NIC-04-P-INFRAADM »
- « NIC-05-V-INFRA »
- « NIC-06-P-VLAN4 »

→ Serveur Dell T630 :

→ Windows Serveur 2012 R2

→ Nom de machine : HV%codehex%10

→ Adresse IP : 10.octet2.octet3.109

(À noter que cette configuration n'est que temporaire puisqu'elle sera remplacée par celle du T610 pour le remplacer après l'exportation)

→ Variables d'environnement des deux serveurs :

- « octet2 » = « 10 »
- « octet3 » = « 11 »
- « codehex » = « 0A0B »

Les variables d'environnement « octet2 » et « octet3 » varient en fonction de l'établissement et le « codehex » (code hexadécimal) est défini par ces dernières.

- Connexion entre les deux serveurs via switch cisco catalyst 2960 (classe 1)
- Environnement de développement : PowerShell Studio 2012

PREPARATION

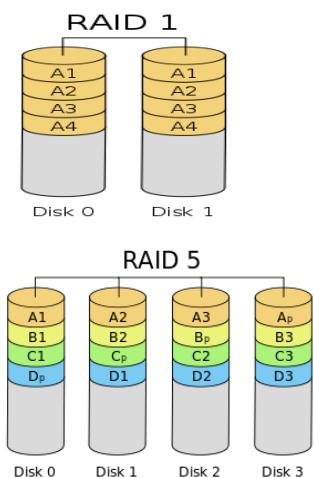
INSTALLATION DES SERVEURS DINDON

MISE EN PLACE, PREPARATION ET CONFIGURATION DU MATERIEL

Après avoir étudié notre cahier des charges, nous avons mis en place notre matériel et configuré celui-ci : mise en place des deux serveurs (le Dell T610 et le Dell T630) et du switch, mise en place des RAID (Redundant Array of Independent Disks), installation des systèmes d'exploitation (Windows Server 2012 R2), installation de Hyper-V, création des machines virtuelles, configurations des cartes réseaux (physiques et virtuelles) et des pare-feux, mise en place des rôles et services (Active Directory, Domain Name System, Dynamic Host Configuration Protocol et Windows Deployment Services)...



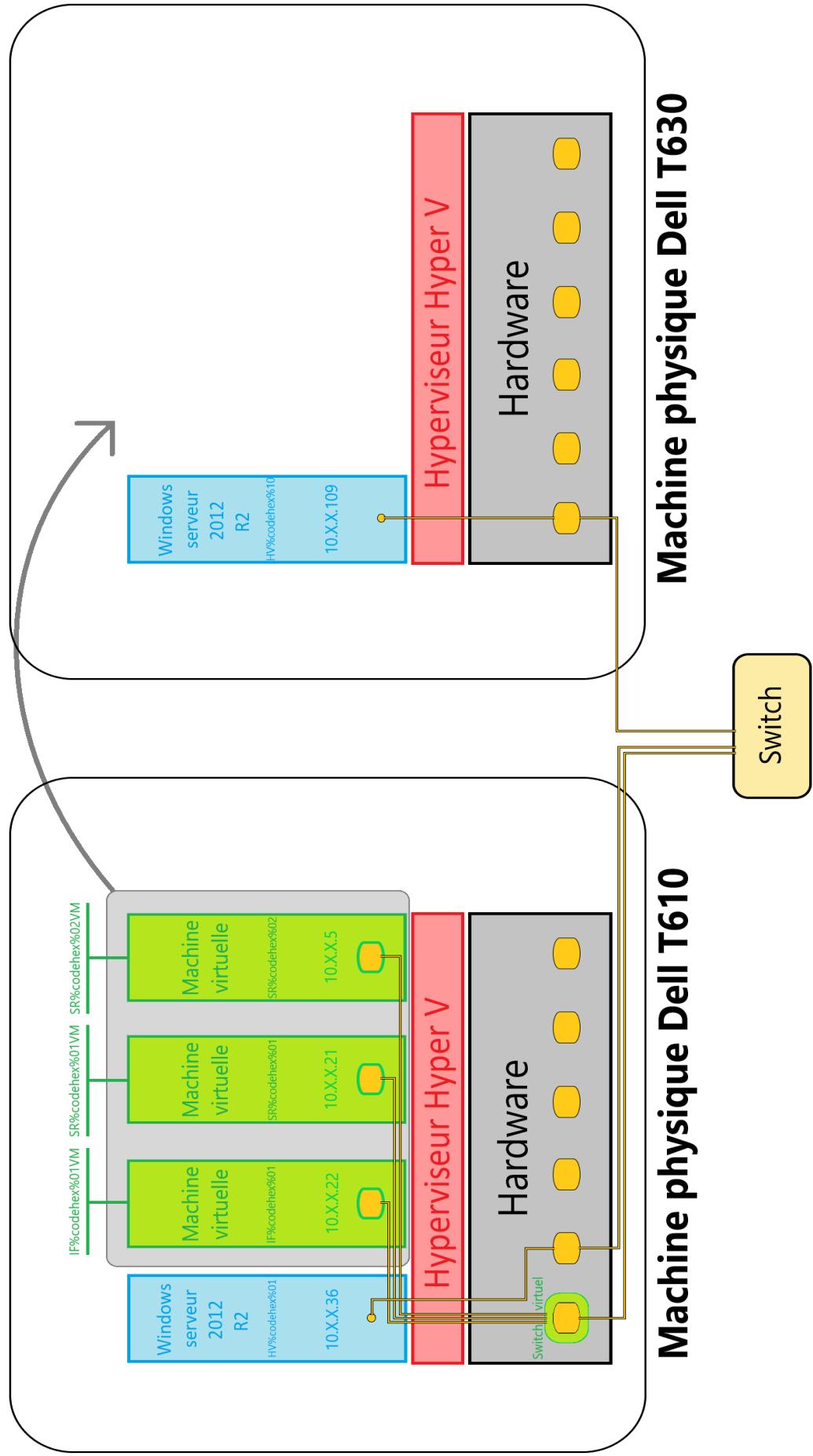
Pendant cette installation nous avons pu rencontrer la notion de RAID puisque nos serveurs Dindons possédaient 5 disques durs chacun, 2 en RAID 1 (écriture des données en miroir) et 3 en RAID 5 (la capacité de stockage d'un disque dur étant perdue, afin d'écrire des bits de parité, permettant de restaurer des données perdues en cas de défaillance).



Une fois le tout mis en place, il ne restait plus qu'à créer le script de migration des machines virtuelles, qui est le travail qui a été défini comme étant notre mission de stage.

Projet Dindon

RESUME DE L'INSTALLATION



PREPARATION AU POWERSHELL

RECHERCHE DE COMMANDES

Pour commencer, n'ayant aucune expérience avec le langage PowerShell, la première directive que nous avons reçue consistait en une recherche des commandes de base, ainsi que des commandes apportées par Hyper-V, qui auraient pu nous servir pour la suite de notre mission. De cette façon nous avons pu acquérir une liste de commandes de base permettant la création du script:

Commandes	Description
New-VM	Créer une machine virtuelle
Remove-VM	Supprimer une machine virtuelle
Start-VM	Démarrer une machine virtuelle
Stop-VM	Arrêter une machine virtuelle
Rename-VM	Renommer une machine virtuelle
Export-VM	Exporter une machine virtuelle
Import-VM	Importer une machine virtuelle
Get-VM	Récupérer l'état actuel d'une machine virtuelle
Test-Connection	Tester la connectivité entre machines
Netsh	Modifier le paramétrage d'une carte réseau
Start-Process	Lancer un exécutable extérieur au script
Net Use	Créer un lecteur réseau
...	...

TESTS DES COMMANDES

Une fois ces quelques commandes notées, et après avoir créé quelques machines virtuelles, nous avons pu tester le fonctionnement et vérifier la syntaxe de chacune de ces commandes de manière à être aptes à commencer la création du script, que notre tuteur nous a vivement conseillé de produire étape par étape, en fonction des tâches qu'il avait à accomplir.

CREATION DU SCRIPT

SERVEUR DINDON T610

ARRET DES MACHINES VIRTUELLES

Pour commencer, le script doit éteindre les trois machines virtuelles, pour ce faire on utilise la commande « Stop-VM ».

Nous avons au préalable créé une fonction « etat-vm » retournant l'état actuel de la machine virtuelle traitée dans la boucle « foreach » de notre fonction « arret-vm » (boucle s'exécutant une fois par machine).

```
41
42 function etat-vm ($nomServeur) {
43
44     $paraVM = Get-VM $nomServeur
45     $etatVM = $paraVM.state
46
47     return $etatVM
48 }
```

Cette dernière peut renvoyer de multiples chaînes de caractères, dont les plus courantes sont « Off » (si la machine est arrêtée) et « Running » (si la machine est allumée).

Nous traitons ensuite cette information afin de déterminer s'il faut donner l'ordre à cette machine virtuelle de s'éteindre et mettre à jour l'affichage (disponible en annexes).

```
131
132 $nomServeurs = $nomServeur1, $nomServeur2, $nomServeur3
133
134 foreach ($nomServeur in $nomServeurs)
135 {
136     $i = 0
137
138     while (((etat-vm $nomServeur) -ne "Off") -and ($i -lt 10))
139     {
140         $labelInfo.Text = $nomServeur + " est en cours d'arrêt"
141         Stop-VM $nomServeur
142         $i = $i + 1
143     }
144
145     if ($i -ge 10)
146     {
147         [void][System.Windows.Forms.MessageBox]::Show("Arrêt de " + $nomServeur + " impossible, veuillez contacter le service support")
148     }
149     verif
150 }
151
152 $labelInfo.Text = ""
153
154
```

A noter que les variables « \$script:nomServeur1 », « \$script:nomServeur2 » et « \$script:nomServeur3 » ont été déclarées plus tôt de manière à s'adapter au « codehex » correspondant à l'établissement :

TEST DE LA CONNEXION

Le test de connexion (commande : « Test-Connection ») nous permet, via le protocole ICMPv4, de vérifier la connectivité entre deux machines. En y ajoutant « -Quiet », nous pouvons récupérer cette information dans une variable booléenne (« True » ou « False »).

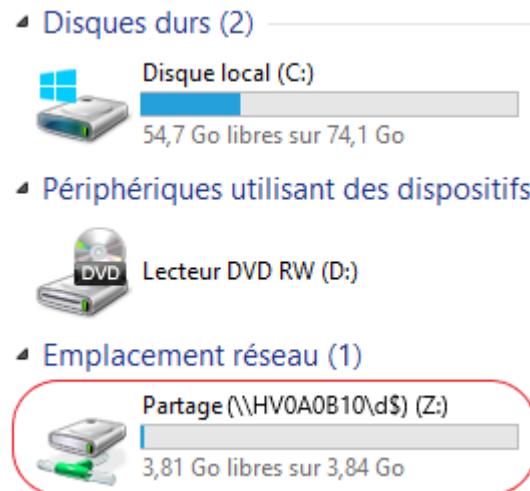
```
157  function testConnexion
158  {
159      $script:funTestConnexion = $false
160
161      $buttonTestConnexion.Text = "Test de connexion en cours..."
162      $buttonTestConnexion.BackColor = 'Gold'
163
164      $etatReseau = Test-Connection 10.$env:octet2.$env:octet3.109 -Quiet
165
166      if ($etatReseau -eq $true)
167      {
168          $buttonTestConnexion.Text = "Connexion réussie"
169          $buttonTestConnexion.Backcolor = 'Chartreuse'
170          $buttonTestConnexion.Enabled = $false
171          $script:funTestConnexion = $true
172
173      }
174      else
175      {
176          $buttonTestConnexion.Text = "Aucune connexion Reconnexion ?"
177          $buttonTestConnexion.Backcolor = 'red'
178      }
179  }
```

Nous avons fait en sorte que ce test soit nécessaire à la suite du script car si la connexion n'est pas établie, le bouton permettant d'effectuer le mappage réseau ne sera pas cliquable (puisque un mappage réseau ne peut évidemment s'effectuer sans connexion avec l'autre serveur) car il est directement lié à la variable « \$script:funScriptTestConnexion » qui permet l'activation du bouton.

Le préfix « script: » permet à une variable déclarée dans une fonction d'étendre sa portée à l'intégralité du script.

MISE EN PLACE DU MAPPAGE RESEAU

Le mappage réseau consiste en la mise en place d'un lecteur réseau. Dans notre cas, nous avons créé un lecteur nommé « Z », mappé sur le lecteur « D » du second serveur.



Pour ce faire, nous nous sommes servis de la commande « net use ».

```
184     function MappageReseau
185     {
186         try
187         {
188             $buttonMappage.BackColor = 'gold'
189             $buttonMappage.Text = "Mappage en cours ..."
190
191             $user = ("HV" + $env:codehex + "01\Administrateur")
192             net use Z: \\10.$env:octet2.$env:octet3.109\d$ /user:$user "P@ssword1"
193
194
195             $buttonMappage.Enabled = $false
196             $buttonMappage.BackColor = 'chartreuse'
197             $buttonMappage.Text = "Mappage terminé."
198             $script:funMappage = $true
199
200
201         catch
202         {
203             $buttonMappage.Backcolor = 'red'
204             $buttonMappage.Text = "Echec du mappage Réessayer ?"
205             [void][System.Windows.Forms.MessageBox]::Show("Echec du mappage réseau.", "ERREUR")
206         }
207     }
208 }
```

Le signe « \$ » après le nom du lecteur mappé permet de spécifier que ce partage n'est visible et disponible que pour les comptes ayant les droits administrateurs.

EXPORTATION DES MACHINES VIRTUELLES

L'exportation des machines virtuelles se fait vers le lecteur mappé sans nécessité de copie préalable. Cette exportation s'effectue via la commande « Export-VM ». De la même façon que pour la fonction « arret-vm », notre fonction « export-vm » se construit autour d'une boucle « foreach » pour donner l'ordre d'exportation une machine virtuelle après l'autre.

La commande principale de cette fonction permet donc d'exporter la machine virtuelle actuellement traitée par la boucle « foreach » vers le lecteur réseau.

Export-VM \$nomServeur -Path Z:

A partir de cette étape du script, nous avons séparé notre script en deux, en créant un exécutable pour chaque serveur. L'ancien serveur n'ayant plus rien à accomplir, nous pouvons l'éteindre définitivement grâce à la commande suivante :
« Shutdown -t 0 -Force »

SERVEUR DINDON T630

IMPORTATION DES MACHINES VIRTUELLES

L'importation est effectuée d'une manière très similaire à l'exportation, grâce à la commande « export-VM » dans une boucle « foreach », permettant d'importer les machines virtuelles une à une depuis le disque D (disque sur lequel nous avons préalablement exporté les machines virtuelles).

CONFIGURATION

Pour finir, il ne reste plus qu'à reprendre la configuration de l'ancien serveur en prenant son nom de machine et sa configuration réseau : adresse IP, masque, DNS par défaut, et domaine :

```
25 function newConfig
26 {
27     Rename-Computer -ComputerName ("HV" + $env:codehex" + "10") -NewName ("HV" + $env:codehex" + "10")
28     netsh interface ipv4 set address name= "NIC-02-INFRA" static ("10." + $env:octet2 + "." + $env:octet3 + ".36") 255.255.255.0
29     Set-DnsClientServerAddress -InterfaceIndex 12 -ServerAdresses ("10." + $env:octet2 + "." + $env:octet3 + ".22")
30     Add-Computer -DomainName ("PDV" + $env:codehex + ".mousquetaires.local") -Credential ("Administrateur@PDV" + $env:codehex + ".mousquetaires.local")
31     $boutonIP.BackColor = "Chartreuse"
32     $boutonIP.Text = "Changement effectué"
33     $boutonIP.Enabled = $false
34 }
```

Puis en la redémarrant pour confirmer les modifications :

Shutdown -t 0 -Force -r

Une fois ceci fait, le serveur est opérationnel, il ne reste plus qu'à lancer les machines virtuelles pour que ce nouveau serveur Dindon prenne le relai.

LIMITES ET PROBLEMES RENCONTRES

MAPPAGE

Durant la phase de développement de la partie « mappage » de notre script, nous avons dû faire face à un problème. En effet, même après le mappage réseau effectué, il nous était impossible d'exporter les machines virtuelles vers notre nouveau lecteur Z (apparition d'un simple message d'erreur nous avertissant de l'échec de l'exportation). Notre tuteur nous a alors expliqué qu'il avait aussi rencontré cette erreur lors du développement de son script il y a quatre ans. Le problème peut être contourné en effectuant la commande « net use », via « PSEexec », un outil nous permettant d'exécuter la commande en utilisant les droits administrateur d'un compte d'une machine distante (ici le Dell T630).

THREAD

Notre tuteur nous a fait remarquer que notre script possédait une limite importante. Notre script ne pouvait exécuter deux tâches en même temps. Pendant l'exportation des machines virtuelles par exemple, notre script étant déjà occupé, l'interface graphique n'est plus gérée : la fenêtre ne peut plus être déplacée, les boutons ne peuvent plus être appuyés etc...

Une solution à ce problème aurait été la gestion des tâches à exécuter par un « Thread », permettant de traiter l'interface et les processus extérieurs en parallèle.

LOGS

Une seconde limite à notre script est qu'il ne possède pas de moyen de sauvegarde. Dans le cas où le technicien fermerait la fenêtre avant d'avoir terminé, le script une fois relancé, repartirait du début, et n'aurait pas pris en compte que le mappage réseau avait déjà été effectué (par exemple).

Nous aurions pu utiliser des « logs », des fichiers créés à chaque étape de notre script (avec la commande « New-Item »), permettant à chaque exécution de déterminer quelles sont les étapes qui ont déjà été effectuées, après vérification d'existence de ces fichiers (avec la commande « Test-Path »).

CONCLUSION

Mes six semaines se sont passées dans une ambiance à la fois très agréable et très professionnelle, les personnes nous ayant pris en charge ont toutes été extrêmement présentes et patientes avec nous, nous permettant de passer un stage dans d'excellentes conditions. Nous avons pu y découvrir une entreprise, sa structure et son fonctionnement, ainsi que plusieurs notions en informatique, ce qui était bien évidemment ce que je recherchais pour ce stage.

J'ai d'abord pu aborder des sujets sur certaines infrastructures réseau, comme le fonctionnement des parcs informatiques de grands établissements. Dans mon cas, j'ai eu l'occasion d'étudier ceux de supermarchés d'enseignes majeures.

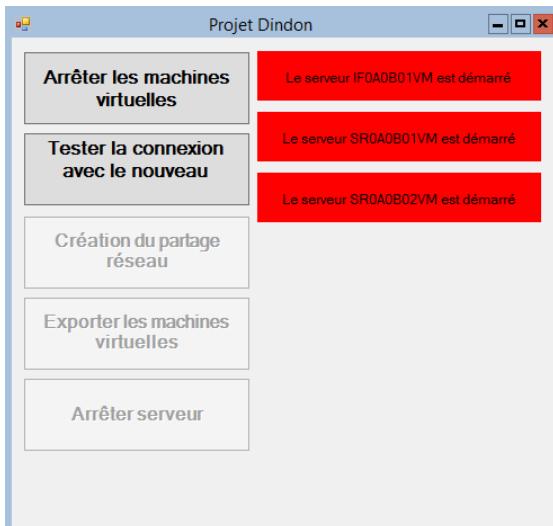
J'ai aussi pu découvrir le langage PowerShell de Microsoft, un langage de script qui permet l'enchaînement de commandes systèmes de manière logique, permettant l'automatisation totale de tâches complexes et fastidieuses à faire manuellement.

En parallèle de ce langage, j'ai aussi découvert la virtualisation, son fonctionnement et ses applications concrètes, me faisant prendre conscience des capacités que cela présente.

Pour résumer, ce stage fut pour moi une occasion de côtoyer pendant six semaines une équipe aussi chaleureuse que compétente professionnellement, auprès de laquelle j'ai beaucoup appris. Ces multiples notions portants sur le domaine de l'informatique m'auront permis d'en découvrir plus sur cet univers et me seront sans doute indispensables pour la suite de mon parcours.

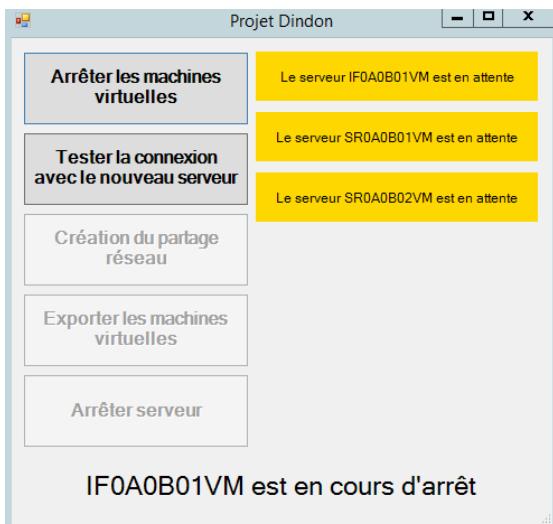
ANNEXES

Fonctionnement d'une étape du script : fonction « arret-vm » :



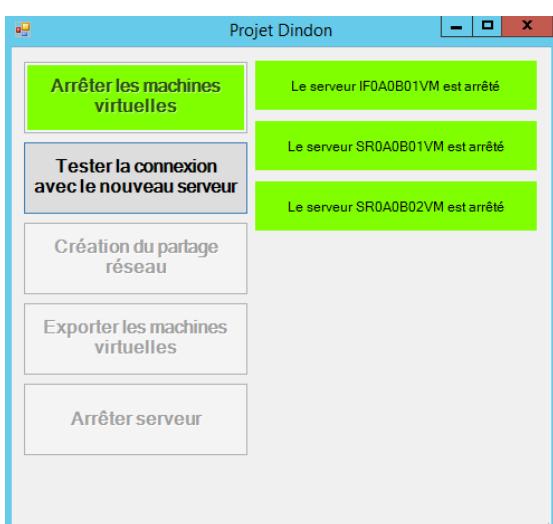
Lancement du script :

Les machines virtuelles encore démarré sont marquées dans un cadre rouge. Si au moins une machine est démarrée, alors le bouton « Arrêter » est cliquable.



Pendant l'arrêt :

Les machines virtuelles en cours d'arrêt sont marquées dans un cadre jaune. L'ordre d'arrêt est donnée une machine après l'autre.

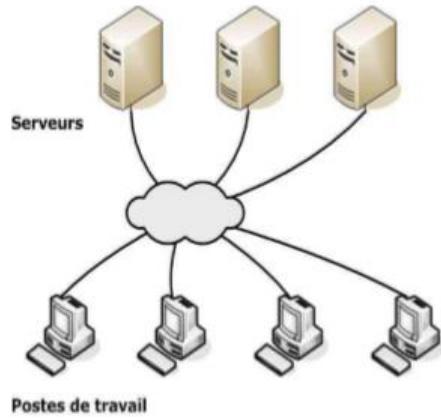


Après l'arrêt :

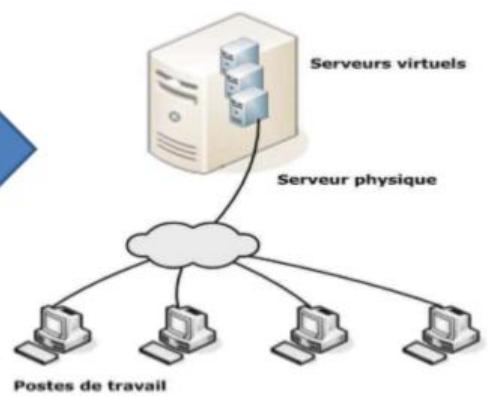
Les machines virtuelles arrêtées sont marquées dans un cadre vert. Si les machines se sont arrêtées ou qu'elles l'étaient déjà avant le lancement du script, alors le bouton d'arrêt n'est pas cliquable.

Fonctionnement des machines virtuelles :

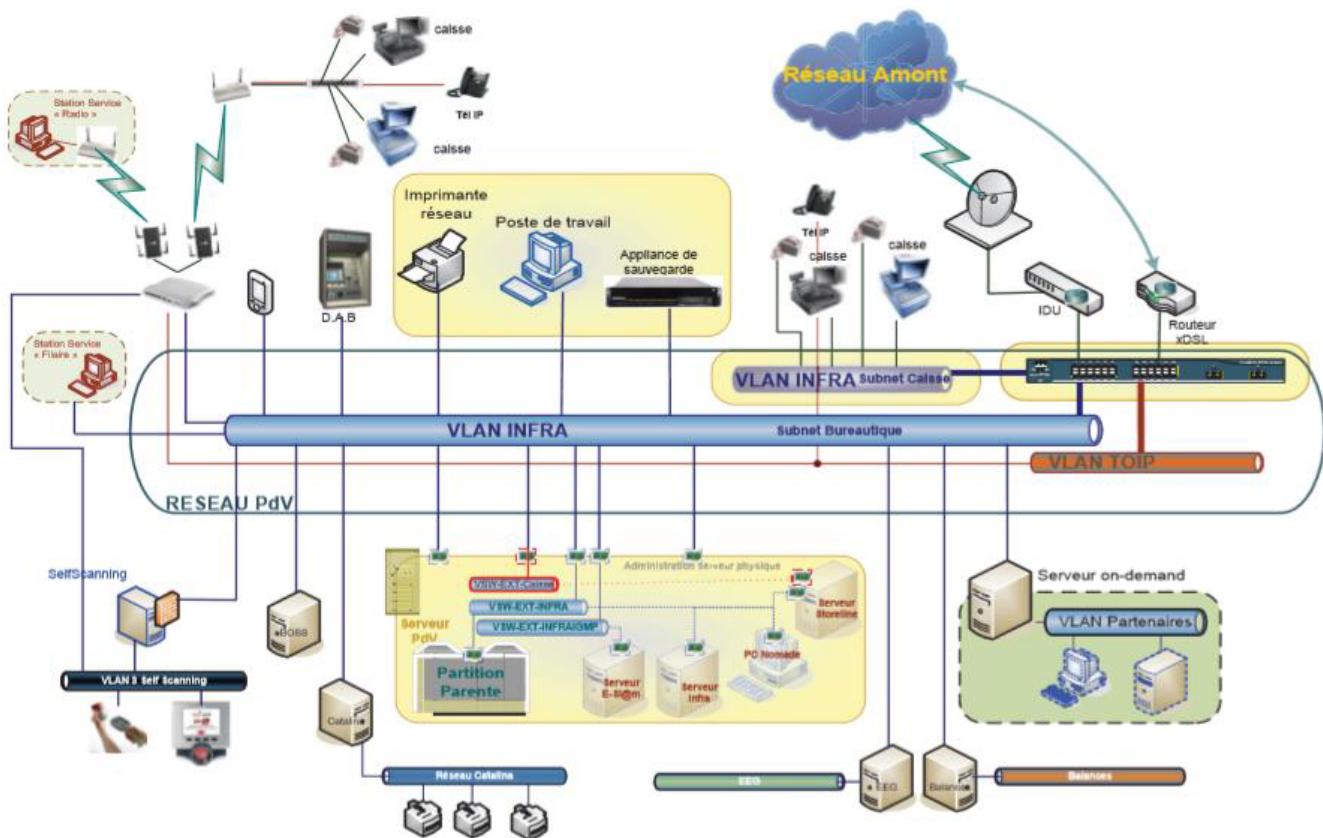
Architecture Classique Esiam



Architecture Virtualisé Dindon



Architecture du réseau des établissements du groupe Les Mousquetaires :



VIRTUAL MACHINE MIGRATION (VM MIGRATION)

DEFINITION - WHAT DOES *VIRTUAL MACHINE MIGRATION* (*VM MIGRATION*) MEAN?

Virtual machine migration is the task of moving a virtual machine from one physical hardware environment to another. It is part of managing hardware virtualization systems and is something that providers look at as they offer virtualization services.

Virtual machine migration is also known as teleportation.

TECHOPEDIA EXPLAINS *VIRTUAL MACHINE MIGRATION* (*VM MIGRATION*)

In hardware virtualization, physical hardware pieces are carved up into a set of virtual machines logical hardware pieces that do not have a physical shell or composition, which are essentially just programmed pieces of an overall hardware system. In a virtualization setup, a central hypervisor or another tool allocates resources like CPU and memory to virtual machines. For instance, in older networks, most of the individual elements were physical workstations, such as desktop PCs, which were connected by Ethernet cabling or other physical connections. By contrast, virtual machines do not have a physical interface. They do not have a box or shell or anything to move around. But they can be connected to the same keyboards, monitors and peripherals that humans have always used to interact with personal computers.

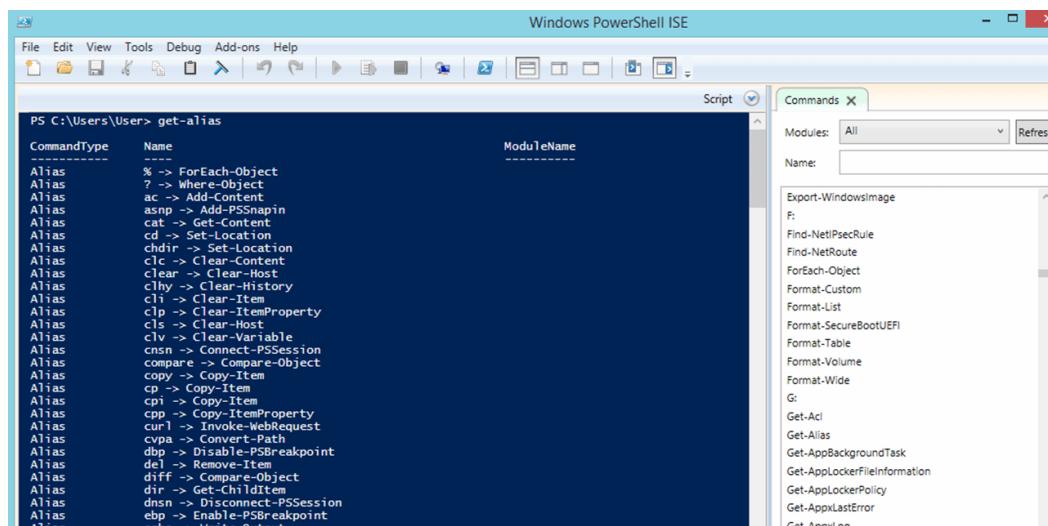
In virtual machine migration, system administrators move these virtual pieces between physical servers or other hardware pieces. In an effort to facilitate this, a new kind of migration has evolved called "live virtual machine migration." Live migration involves moving these virtual machines without shutting down a client system. Modern services often provide live migration functionality to make it easier to move virtual machines without doing a lot of other administrative work.

Source : www.techopedia.com/definition/15033/virtual-machine-migration-vm-migration

PowerShell Integrated Scripting Environment

PowerShell Integrated Scripting Environment (ISE), introduced by Microsoft in PowerShell version 2.0, is a PowerShell host application used to write, test and debug scripts or write commands in a Windows GUI.

PowerShell ISE comes with multiple features, such as syntax coloring, multiline editing, context-sensitive help and tab completion.



PowerShell ISE has sophisticated features that are familiar to Windows users. For instance, a user can highlight and copy a portion of a PowerShell command with a mouse or with the Shift + Arrow hotkey combination. The user can also paste the content anywhere in the editor window.

Another useful feature is the ability to keep different versions of a command in the editor and run commands that you need in the PowerShell ISE.

The F5 key launches a command directly from the editor. To execute a particular line, select it and press F8. The context-sensitive help displays matching cmdlets when the user starts to enter a command. A command add-on shows a list of cmdlets to select.

PowerShell ISE provides tabs to allow work on multiple administrative tasks. PowerShell ISE enables quick switching from the CLI to scripting mode.

LANGUAGE CONSTRUCTS

As a scripting language, PowerShell offers several language constructs that control the flow of scripts, while making decisions as to what should be done. Some of the language constructs include conditionals, switches, loops and variables.

HYPERVERISOR DEFINITION

A hypervisor is a process that separates a computer's operating system and applications from the underlying physical hardware. Usually done as software although embedded hypervisors can be created for things like mobile devices.

The hypervisor drives the concept of virtualization by allowing the physical host machine to operate multiple virtual machines as guests to help maximize the effective use of computing resources such as memory, network bandwidth and CPU cycles.

BENEFITS OF HYPERVISORS

Even though VMs can run on the same physical hardware, they are still logically separated from each other. This means that if one VM experiences an error, crash or a malware attack, it doesn't extend to other VMs on the same machine, or even other machines.

VMs are also very mobile – because they are independent of the underlying hardware, they can be moved or migrated between local or remote virtualized servers a whole lot easier than traditional applications that are tied to physical hardware.

There are two types of hypervisors, creatively named Type 1 or Type 2. Type 1 hypervisors, sometimes called “native” or “bare metal” hypervisors, run directly on the host’s hardware to control the hardware and manage the guest VMs. Modern hypervisors include Xen, Oracle VM Server for SPARC, Oracle VM Server for x86, Microsoft Hyper-V and VMware’s ESX/ESXi.

Type 2 hypervisors, sometimes called “hosted hypervisors,” run on a conventional OS, just like other applications on the system. In this case, a guest OS runs as a process on the host, while the hypervisors separate the guest OS from the host OS. Examples of Type 2 hypervisors include VMware Workstation, VMware Player, VirtualBox and Parallels Desktop for Mac.

In the enterprise data center space, consolidation has resulted in three major vendors on the hypervisor front: VMware, Microsoft and Citrix Systems.

Source : www.networkworld.com/article/3243262/virtualization/what-is-a-hypervisor.html

Nom de l'établissement Lycée Vauvenargues 60, boulevard Carnot 13100 Aix-en-Provence Tel : 04-42-17-40-57 Mail : cdttaix@gmail.com	BTS Systèmes Numériques	Session 2019
STAGE EN MILIEU PROFESSIONNEL		
CERTIFICAT DE STAGE <i>(à faire figurer dans le rapport de stage)</i>		

Nom et prénom du stagiaire : RUIZ MARINO Guillaume

Raison sociale de l'entreprise : ODEOLIS

Nom du responsable de l'entreprise : G. DE LA HORIE

Service d'accueil du stagiaire : SUPPORT TECHNIQUE

Nom du tuteur: FRIEDRICH LAURENT Fonction: SUPPORT TECHNIQUE

N° de tél. : 06 62 16 78 00

Mèl : LAURENT.FRIEDRICH @ ODEOLIS.FR

Dates de début et de fin du stage : 14 mai – 22 juin 2018

Nombre de ½ journée(s) d'absence excusée(s) : / non excusée(s) : /

Activités conduites par le stagiaire pendant le stage :

PROJET DE MIGRATION DE SERVEUR HYPER-V
DEVELOPPEMENT D'UN OUTIL EN POWERSHELL.

Appréciation générale du tuteur sur le stagiaire :

STAGIAIRE MOTIVE, INTERRESSE ET AUTONOME

