Process de création de VM & paramétrages additionnels

---

1. Création de VMs via la plateforme MonInfra 1 1.1 Création d’une VM de TEST 1 1.2 Création d’une VM de PROD 2 2. Tags sur la VM 2 3. Activation du SSH 2 4. Copier/coller dans VIM 2 5. Créer le user admin-sig 3 6. Associer /bin/bash à admin-sig 3

```
7. Permettre les ssh pour admin-sig 4
```

9. Permettre le ssh entre la VM de test et la VM de prod 4 10. Ajouter clés SSH 5 11. Option – modif specs VM 5 12. Scripts de déploiement 5 13. Installer docker et admin-sig 5 14. Installer l'application 6 15. Paramétrer la redirection 6

---

1. Création de VMs via la plateforme MonInfra On se connecte à l’URL suivante : https://moninfra.intra.univ-nantes.fr/accounts/login/ 1.1 Création d’une VM de TEST

```
VM de TEST
  choisir PVE-Test
  Debian S - 5GO
  nom instance : !!! IMPORTANT ne pas mettre "test" dans le nom !!!!
  champ "comments"
    demander de mettre 4GO de RAM car sinon 1GO...
  fqdn : sera de la forme <nomapplication>.test.univ-nantes.prive
```

1.2 Création d’une VM de PROD

```
  choisir PVE-Production
  Debian - [M] ou Debian - [L]
  dans le champ "comments"
    VM pour héberger l'instance esup-stage.
  fqdn : sera de la forme <nomapplication>-dprv.univ-nantes.prive
  ps : pour les 3e 1e VM on était pas passé par MonInfra mais par le ticket
Canum n° 46577
```

2. Tags sur la VM Proxmox : aller mettre les tags sur la VM

1. medium
2. prod

3. Activation du SSH se connecter en ssh en ajoutant un alias sur son .basrhc local

```
alias stage-test='ssh root@stage-test.test.univ-nantes.prive'
tester la connexion en SSH
tester le scp : scp fic.txt root@stage-test.test.univ-nantes.prive:/tmp
```

4. Copier/coller dans VIM permettre le copier/coller dans vim

```
echo 'set mouse=' >> ~/.vimrc
```

copier les alias .bashrc de la VM source vers la nouvelle VM

installer telnet, net-tools, dnsutils

```
apt update
apt install xinetd telnetd telnet -y
systemctl status inetd
apt install net-tools netcat-traditional -y dnsutils
```

autoriser les flux vers les bases de donnés (canum 57612, 60779)

```
Lien du formulaire CANUM :
https://canum.univ-nantes.fr/marketplace/formcreator/front/formdisplay.php?id=33
Sources : Depuis quel(s) réseau(x) de l'UN, quel(s) serveur(s) (FQDN ou a défaut IP) : pegase.dprv.univ-nantes.prive
Destinations : Vers quel(s) réseau(x) de l'UN, quel(s) serveur(s) (FQDN ou a défaut IP) : ditto.sig.univ-nantes.prive
Services : Quel(s) service(s), port TCP/UDP, protocole(s) : Base PostgreSQL, port 5432
```

```
tester si les flux sont ouverts :
  nc -zvw3 enton.sig.univ-nantes.prive 3306
  nc -zvw3 omot.sig.univ-nantes.prive 3306
  nc -zvw3 bdd-geode.presidence.univ-nantes.prive 1521
si non ouvert --> dans proxmox firewall ajouter les groupes sig-mariadb-out et sig-oracle-out
```

5. Créer le user admin-sig

```
useradd -m admin-sig -s /bin/bash
passwd admin-sig  --> Ll....
```

6. Associer /bin/bash à admin-sig Par défaut c’est /bin/sh

```
  # --- À exécuter en root ---
  # Mettre bash comme shell par défaut pour l’utilisateur admin-sig (une seule fois)
  usermod -s /bin/bash admin-sig
  # S’assurer que /bin/bash est listé comme shell autorisé
  grep -qx '/bin/bash' /etc/shells || echo /bin/bash >> /etc/shells
  # --- Basculer sur l’utilisateur ---
  su - admin-sig
  # Vérifier que l’on est bien passé sur le bon utilisateur et sur bash
  whoami            # attendu : admin-sig
  echo "$SHELL"     # attendu : /bin/bash
  ps -p $$ -o comm= # attendu : bash
```

```
7. Permettre les ssh pour admin-sig
  # --- Créer ~/.ssh pour admin-sig et ajouter la clé publique ---
  # 1) Créer le dossier et régler lescd permissions
  mkdir -p ~/.ssh && chmod 700 ~/.ssh
  # 2) S'assurer que le fichier existe avec les bons droits
  touch ~/.ssh/authorized_keys && chmod 600 ~/.ssh/authorized_keys
  # 3) Ajouter les clés publiques dans ~/.ssh/authorized_keys
```

8. Tester le SFTP via filezilla

```
avec le user admin-sig type : SFTP authentification : normale
```

9. Permettre le ssh entre la VM de test et la VM de prod

```
sera également un backup au cas où notre clé privée ne fonctionnerait plus
ssh-keygen -t ed25519
mot  depasse : Ll....
cat stage.pub >> authorized_keys  IMPORTANT : bien mettre >> et non pas >
tester : scp f.txt root@stage-test@test.univ-nantes.prive:/tmp
--> ne fonctionne pas entre dprv.univ-nantes.fr et test.univ-nantes.prive
car le réseau est bloqué
--> il faudrait faire un ticket
```

10. Ajouter clés SSH

1. → plus besoin car c'est géré dans installation.sh

collègues doublons DV, FA et FP

```
jenkins : vulpix
cat albert-f.pub >> authorized_keys
```

11. Option – modif specs VM augmenter la RAM / augmenter l'espace disque / demander des partitions

```
il faut passer par un ticket CANUM à support-infra-système-stockage pour
toute action sur la VM
exemple pour augmenter la RAM :
raison : taille S = 1GO par défaut...
vérifier taille : free -h
intitulé ticket : VM #108 stagetest - augmentation RAM
groupe : support-infra-système-stockage
catégorie : demande / exploitation / système
exemple : ticket 58365
prise en compte de l'augmentation :
  shutdown -h now (-h car à chaud ne suffit pas)
  redémarrer la VM sur le PVE de test de proxmox (on a pas la main il faut
demander à l'infra)
```

12. Scripts de déploiement copier les scripts de déploiement de la VM source vers la VM cible

```
mkdir /usr/local/scripts
copier les scripts de la VM source dedans
```

13. Installer docker et admin-sig

```
utiliser script 'installation.sh' hébergé sur gitlab projet "scripts" (ou
alors manuellement cf. en bas de ce fichier):
```

14. Installer l'application

```
si déjà dockerisée, déplacer de la VM source vers la VM cible
  VM source kakuna : créer un tar gz du répertoire esup-stage
    cd /docker
    tar -czvf esup-stage.tar.gz --exclude='esup-stage/ROOT.war' esup-stage/
  VM cible :
    transferer le tar.gz en local : scp root@stage-test.test.univ-
nantes.prive:/docker/esup-stage.tar.gz .
    copier du local vers la VM : scp esup-stage.tar.gz root@stage-
test.test.univ-nantes.prive:/docker
    tar xvzf esup-stage.tar.gz
    chown -R admin-sig:admin-sig esup-stage/
sinon créer de toute pièce via des conteneurs dockers
```

15. Paramétrer la redirection

```
cf paragraphe "Paramétrer la redirection" plus bas dans le fichier.
```

modifier la redirection de l'url dans le DNS vérifier la redirection

```
  curl -v https://test-esupstage.intra.univ-nantes.fr
  dig test-esupstage.intra.univ-nantes.fr
  traceroute test-esupstage.intra.univ-nantes.fr
```

```
créer ticket CANUM (exemple test : 58855 prod : 50500, 63691, création
nouvelle : 64528)
    formulaire 'ticket technicien'
    catégorie : demande / exploitation / système
    groupe : support-infra-système-stockage
    nom ticket : Mise à jour redirection <application>
    texte :
 Bonjour,
   Pourriez-vous modifier la redirection de l'application
https://application.intra.univ-nantes.fr ?
   Nouvelle redirection :
 VM : application.dprv.univ-nantes.prive
```

```
    redirection en PROD (exemple : 50500)
      prévoir l'heure de migration
```

```
    jour J (spécifique esup-stage)
```

```
      - redéployer le /datas de galopa vers stage (cf
maintenance_estage.doc)
      - arrêter stage sur la VM galopa : docker compose down
      - démarrer stage sur la VM stage : docker compose up -d --build
      - attendre la redirection
      - tester url
      - vérifier logs
      - faire une convention
      - signer 100%
      - supprimer convention
```

arrêter l'ancien service sur la VM source

```
a2dissite stage
arrêter le service docker : docker compose down
arrêter les redémarrages automatiques
  docker-compose.yml
    commenter --restart unless-stopped
  docker-sun.sh
    docker run -d -v ./logs:/logs -p 9090:9090  --restart unless-stopped --
name="esup-siscol" local/esup-siscol:v1.0.28
    devient
    docker run -d -v ./logs:/logs -p 9090:9090  --name="esup-siscol"
local/esup-siscol:v1.0.28
```

fonctionnement validé avec

```
  ip route list
    default via 172.20.99.1 dev eth0 proto static
    172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1
    172.18.0.0/16 dev br-fbed509dbf0b proto kernel scope link src 172.18.0.1
    172.20.99.0/24 dev eth0 proto kernel scope link src 172.20.99.109
  docker container ps -a
    CONTAINER ID   IMAGE                        COMMAND
CREATED         STATUS             PORTS                                NAMES
    bd64ade27570   estage:latest                "catalina.sh run"        7
minutes ago   Up 7 minutes       8080/tcp, 0.0.0.0:8082->8082/tcp   estage
    59562237886c   local/esup-siscol:v1.0.28   "java -Duser.timezon…"   4
weeks ago     Up About an hour   0.0.0.0:9090->9090/tcp             esup-
siscol
```

Paramétrer la redirection * Les redirections ne peuvent se faire que sur les ports 80 ou 443 En test on ne passe pas par le reverse proxy mais par ALOHA, il nous faut des certificats SSL si on veut faire du HTTPS En prod on passe par le reverse proxy donc pas besoin de certificats pour faire du HTTPS

Redirection sur une VM de TEST

```
http://nomapp.test.univ-nantes.prive est automatiquement créé à la création
de la VM --> accessible uniquemnt dans notre réseau pas par les composantes
```

```
les Vms de tests sont accessibles uniquement par la DSIN
Idéalement il faudrait plutôt des VMs de préprod pour ques les gestionnaires
aient accès.
Donc pour le moment demander des urls de préprod vers les VM de tests en
attendant la futur automatisation sur proxmox
--> https://<nomapplication>.preprod.intra.univ-nantes.fr

Redirection via traefik au lieu d'apache (port 80)
  traefik est un reverse proxy qui permet de faire comme apache
  cf VM contact.test.univ-nantes.prive fichier /docker/contact/docker-
compose.yml

Redirection via traefik au lieu d'apache (port 443 avec certificats sur la
VM)
  cf VM stage-test.univ-nantes.prive fichier /docker/esup-stage/docker-
compose.yml
Redirection via aucun reverse proxy pour du HTTP sur du 80
  pas besoin d'installer APACHE
  docker-compose.yml --> changer le port pour passer sur du 80
    ports:
        - "80:8082" 80 : machine locale / 8082 : conteneur
```

```
  initialement (kakuna)
        - "8082:8082"
  application.properties
     appli.url=https://test-esupstage.intra.univ-nantes.fr/frontend/#/
     appli.prefix=https://test-esupstage.intra.univ-nantes.fr
     initialement (kakuna)
       appli.url=https://test-esupstage.intra.univ-nantes.fr/frontend/#/
       appli.prefix=https://test-esupstage.intra.univ-nantes.fr
   nudir : https:// devient http(s)?://

Redirection via Apache pour du HTTPS sur du 443
  Installation apache (2.4.65)
    idéalement dans un conteneur docker mais pour le moment via apt
    apt update && sudo apt upgrade -y
    apt install apache2
    systemctl start apache2
    docker-compose.yml   --> 8082:8082
    apache2ctl configtest --> tester config
    systemctl enable apache2
    systemctl status apache2
    a2enmod ssl
    a2enmod headers
    a2enmod proxy
    a2enmod proxy_http
    apache2ctl -M --> vérifier modules

  Config apache
    demander certificats SSL intra.univ-nantes.fr à Vincent Bruneton
(23/10/25 : récupérés dans Pole FVE / Système)
```

```
    un seul certificat suffit : le full chain
    renommer les certificats
      fullchain2.pem -->  fullchain.intra.univ-nantes.fr.pem
      privkey2.pem --> privkey2.intra.univ-nantes.fr.pem
    les copier sur la VM
      scp fullchain2.pem root@stage-test.test.univ-nantes.prive:/tmp
      scp privkey2.pem root@stage-test.test.univ-nantes.prive:/tmp
      cd /tmp
      mv fullchain2.pem /etc/ssl/certs/fullchain.intra.univ-nantes.fr.pem
      mv privkey2.pem /etc/ssl/private/privkey2.intra.univ-nantes.fr.pem
    créer le Vhost
      créer le fichier de conf du Vhost dans /etc/apache2/sites-available
      copier un nomAppli.conf d'une autre VM et l'adapter
        SSLCertificateFile   --> certificat full chain
        SSLCertificateKeyFile --> clé privée
      faire du reverse proxy pour rediriger les requêtes 443 vers bnotre
service docker (ex : 8082)
        ProxyPass / http://localhost:8082/
        ProxyPassReverse / http://localhost:8082
    a2ensite estage.conf
```

```
  Modif application.properties
     appli.url=https://test-esupstage.intra.univ-nantes.fr/frontend/#/
     appli.prefix=https://test-esupstage.intra.univ-nantes.fr
```

```
  Modif docker-compose.yml
    -> 80 changer en 8082 sinon conflit

  Restart app
    docker compose up -d --build
```

Redirection sur une VM de PROD

```
Redirection via traefik au lieu d'apache
  cf VM contact.dprv.univ-nantes.prive fichier /docker/contact/docker-
compose.yml
  on reste sur du port 80 car c'est le reverse proxy de l'UN qui gère les
certificats
  docker-compose.yml
    ...
    - "--entrypoints.web.address=:80"
    ports:
    - "80:80"
```

```
    labels:
    - "traefik.enable=true"
    - "traefik.http.routers.contact.rule=Host(`contact.univ-nantes.fr`)"
    - "traefik.http.routers.contact.entrypoints=web"
    - "traefik.http.services.contact.loadbalancer.server.port=9091"
```

```
Redirection via Apache pour du HTTPS sur du 443
Installation apache (2.4.65)
  idéalement dans un conteneur docker mais pour le moment via apt
  apt update && sudo apt upgrade -y
  apt install apache2
  systemctl start apache2
  docker-compose.yml   --> 8082:8082
  tester config
    apache2ctl configtest
  systemctl enable apache2
  systemctl status apache2
  a2enmod ssl
  a2enmod headers
  a2enmod proxy
  a2enmod proxy_http
  apache2ctl -M --> vérifier modules
Config apache
  créer le Vhost en copiant un fichier nomAppli.conf d'une application de
prod
  adapter le fichier conf
     faire du reverse proxy pour rediriger les requêtes 443 vers bnotre
service docker (ex : 8082)
       ProxyPass / http://localhost:8082/
       ProxyPassReverse / http://localhost:8082
  a2ensite estage.conf
  apache2ctl configtest --> tester config
```

```
Tester config apache
  wget --no-proxy http://stage.univ-nantes.fr:80/frontend
  aller voir dans les logs d'apache si on a bien des données
```

Installer Docker manuellement *

```
installer docker (700 MO)
 https://docs.vultr.com/how-to-install-docker-on-debian-12
  docker --version
    Docker version 28.4.0, build d8eb465
  docker compose version
    Docker Compose version v2.39.4
  /!\ En cas de problème d'installation de la clé GPG publique de Docker
  sudo http_proxy=$http_proxy https_proxy=$https_proxy curl -fsSL
https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
```

```
associer admin-sig au groupe docker
  il sera l'utilisateur de docker
  usermod -aG docker admin-sig --> affectation au groupe docker
  cd /
   mkdir /docker --> les applications dockers seront dans ce dosser
  chown -R admin-sig:admin-sig docker/

ajouter le proxy pour docker
```

```
mkdir -p /etc/systemd/system/docker.service.d
vi /etc/systemd/system/docker.service.d/http-proxy.conf

[Service]
Environment="HTTP_PROXY=proxy-upgrade.univ-nantes.prive:3128"
Environment="HTTPS_PROXY=proxy-upgrade.univ-nantes.prive:3128"
Environment="NO_PROXY=localhost,127.0.0.1,localaddress,.localdomain.com"
Relancer
systemctl daemon-reload
systemctl restart docker
systemctl status docker
docker system info | grep -i proxy
```

Divers *** Demander à autoriser automatiquement les accès suivants à chaque VM de TEST ou de PROD

```
annuaire.intra.univ-nantes.fr 636
bdd-geode.presidence.univ-nantes.prive 1521
omot.sig.univ-nantes.prive  3306
enton.sig.univ-nantes.prive 3306
ditto.sig.univ-nantes.prive 5432
```

permettre le ssh entre les VM de test et les VM de prod