Installation d'une machine Publik SaaS

Mode d'emploi pour l'installation d'une machine Publik. Il s'agit d'installer la solution sur une seule machine (physique ou virtuelle).

Cependant, sur le canevas de cette documentation,chaque composant peut également être installé sur une machine dédiée : il faut alors ajuster les règles de pare-feu et les connexions AMQP (exercice laissé au lecteur à ce stade).

Pré-requis

Système d'exploitation Debian 7.x (Wheezy) installé en mode minimal.

Sources APT:

```
deb http://httpredir.debian.org/debian/ wheezy main
deb http://security.debian.org/ wheezy/updates main
deb http://ftp.fr.debian.org/debian wheezy-updates main
deb http://ftp.fr.debian.org/debian wheezy-backports main
deb http://www.rabbitmq.com/debian/ testing main
deb http://deb.entrouvert.org/ wheezy main
```

Installation des paquets «couches basses» :

- nginx-full depuis backports (1.9)
- postgresql depuis backports (9.4)
- python-ndg-httpsclient python-pyasn1 (gestion correcte des requêtes HTTPS par requests)

Déclarations DNS

Dans la zone choisie :

```
portail         A     ....
backoffice      CNAME xxx
connexion       CNAME xxx
demarches       CNAME xxx
passerelle      CNAME xxx
hobo            CNAME xxx
```

Note : dans la suite du document, l'installation est faite dans la zone «example.net»

Préparation PostgreSQL, utilisateurs et base

En tant qu'utilisateur postgres:

```
postgres@xxx$ createuser -SDR hobo ; createdb -O hobo hobo
postgres@xxx$ createuser -SDR authentic-multitenant ; createdb -O authentic-multitenant authentic2
_multitenant
postgres@xxx$ createuser -SDR combo ; createdb -O combo combo
postgres@xxx$ createuser -SDR passerelle ; createdb -O passerelle passerelle
postgres@xxx$ createuser -SdR wcs ; createdb -O wcs wcs_demarches_example_net
```

Système de déploiement hobo + rabbitmq

Installation des paquets:

- erlang-nox — attention, depuis les backports : apt-get install -t wheezy-backports erlang-nox
- rabbitmq-server
- hobo

Configuration nginx depuis ce modèle :

```
# nginx template hobo
server {
    listen 443 ssl;
    server_name hobo.*;    # indiquer ici le ou les noms des instances

    ssl_certificate /etc/ssl/certs/cert.pem;    # un certificat couvrant tous les noms
    ssl_certificate_key /etc/ssl/private/cert.key;

    access_log /var/log/nginx/hobo-access.log combined;
    error_log /var/log/nginx/hobo-error.log;

    location ~ ^/static/(.+)$ {
        root /;
        try_files /var/lib/hobo/tenants/$host/static/$1
                  /var/lib/hobo/tnenats/$host/theme/static/$1
                  /var/lib/hobo/collectstatic/$1
                  =404;
    }

    location ~ ^/media/(.+)$ {
        alias /var/lib/combo/tenants/$host/media/$1;
    }

    location / {
        proxy_pass         http://unix:/run/hobo/hobo.sock;
        proxy_set_header   Host $http_host;
        proxy_set_header   X-Forwarded-SSL on;
        proxy_set_header   X-Forwarded-Protocol ssl;
        proxy_set_header   X-Forwarded-Proto https;
        proxy_set_header   X-Real-IP       $remote_addr;
        proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}

# catchall http → https
server {
    listen   80;
    server_name  hobo.*;    # indiquer ici le ou les noms des instances
    access_log  /var/log/nginx/hobo-access.log combined;
    error_log  /var/log/nginx/hobo-error.log;
    return 301 https://$host$request_uri;
}
```

Lancement de hobo (provoque l'initialisation de la base):

```
root@xxx# service hobo restart
```

Configuration de rabbitmq

```
# zcat /usr/share/doc/hobo/rabbitmq.config.example.gz > /etc/rabbitmq/rabbitmq.config
```

Modifications SSL dans /etc/rabbitmq/rabbitmq.config:

```
…
   {ssl_options, [{certfile,             "/etc/ssl/certs/wildcard.dev.entrouvert.org.pem"},
                  {keyfile,              "/etc/ssl/private/wildcard.dev.entrouvert.org.key"},
                  {versions, ['tlsv1.2', 'tlsv1.1']}]}
…
```

Puis relancer le service

```
root@xxx# service rabbitmq-server restart
```

Création d'un utilisateur hobo dans rabbitmq:

```
root@xxx# rabbitmqctl add_user hobo <password>
root@xxx# rabbitmqctl set_permissions hobo ".*" ".*" ".*"
```

Agent de déploiement

Les instances cibles sont sur la même machine, on installe donc l'agent :

- hobo-agent

Configuration dans /etc/hobo-agent/settings.py est laissée telle quelle:

```
BROKER_URL = 'amqp://'
AGENT_HOST_PATTERNS = None
```

Fournisseur d'identités (authentic)

Paquets :

- authentic2-multitenant

Lancement du processus authentic2:

```
# service authentic2-multitenant restart
```

Configuration nginx inspirée de celle de hobo : remplacer «hobo» par «authentic2-multitenant»

Démarches (w.c.s.)

Paquets

- wcs
- wcs-au-quotidien

Désactivation de wcs au profit de wcs-au-quotidien

```
root@xxx# update-rc.d wcs disable
root@xxx# update-rc.d wcs-au-quotidien enable
root@xxx# service wcs stop
root@xxx# service wcs-au-quotidien restart
```

Configuration nginx, sur modèle hobo mais en mode SCGI au lieu de proxy_pass:

```
(...)
        location /qo { alias  /usr/share/wcs/qommon/; }
        location /apache-errors { alias  /usr/share/auquotidien/apache-errors/; }
        location /themes {
            root /;
            try_files /var/lib/wcs-au-quotidien/$host$uri /usr/share/wcs/$uri =404;
        }

        location / {
            include scgi_params;
            scgi_pass localhost:3001;
            scgi_param SCRIPT_NAME '';
            scgi_param PATH_INFO $uri;
            scgi_param HTTPS yes;
        }
(...)
```

Portails citoyen et agent (combo)

Installation du paquet

- combo

Création des sites nginx (modèle hobo, s/hobo/combo/)

Démarrage:

```
root@xxx# service combo restart
```

Hub de services (passerelle)

Installation du paquet

- passerelle

Création un site nginx (modèle hobo, s/hobo/passerelle/)

Démarrage:

```
root@xxx# service passerelle restart
```

Instanciations

Création d'un fichier JSON «recipe»

Créer un fichier «recipe-example.json» qui va décrire les instances à déployer (remplacer les «example») :

```
{
  "variables": {
    "hobo": "hobo.example.net",
    "authentic": "connexion.example.net",
    "combo": "portail.example.net",
    "combo_agent": "backoffice.example.net",
    "passerelle": "passerelle.example.net",
    "wcs": "demarches.example.net"
  },
  "steps": [
    {"create-hobo": {
      "url": "https://${hobo}/"
    }},
    {"create-superuser": {
      "email": "admin@example.net",
      "password": "example"
    }},
    {"create-authentic": {
      "url": "https://${authentic}/",
      "title": "Connexion"
    }},
    {"set-idp": {
    }},
    {"create-combo": {
      "url": "https://${combo}/",
      "title": "Compte usager",
      "template_name": "portal-user"
    }},
    {"create-combo": {
      "url": "https://${combo_agent}/",
      "slug": "portal-agent",
      "title": "Portail agent",
```

```
      "template_name": "portal-agent"
    }},
    {"create-wcs": {
      "url": "https://${wcs}/",
      "title": "Démarches"
    }},
    {"create-passerelle": {
      "url": "https://${passerelle}/",
      "title": "Passerelle"
    }},
    {"set-theme": {
      "theme": "publik"
    }}
  ]
}
```

Lancement du déploiement

```
# sudo -u hobo hobo-manage cook /path/to/recipe-example.json
```

La commande «cook» attend 30 secondes après chaque déploiement : si le déploiement prend trop de temps (machine lente), il faut relancer la commande autant de fois que nécessaire.

Passage en SQL de wcs

Pour l'instant la création d'une instance wcs ne passe pas automatiquement en mode SQL. Pour l'activer :

- ajouter la ligne suivante dans la section [options] de /var/lib/wcs-au-quotidien/demarches.example.net/site-options.cfg :

```
# extrait de /var/lib/wcs-au-quotidien/demarches.example.net/site-options.cfg

[options]
postgresql = true
... autre options ...
```

- puis lancer la conversion :

```
# sudo -u wcs-au-quotidien wcsctl -f /etc/wcs/wcs-au-quotidien.cfg convert-to-sql --dbname=wcs
_demarches_example_net demarches.example.net
```

Finitions (notes)

- Passer wcs en français depuis https://demarches.example.net/backoffice/settings/language
- Ajouter un rôle "Support et Debogage" (afin d'amorcer le provisionning vers wcs) depuis https://connexion.example.net/manage/roles/