

# Les 7 Démons et 1 script

## Sommaire

- [Introduction](#)
  - [Récapitulatif des différents démons](#)
- [Les étapes de démarrage des différents démons de l'installation Shinken](#)
  - [Etape 1: Démarrage des démons Arbiter et Synchronizer](#)
  - [Etape 2: Lecture des fichiers CFG](#)
  - [Etape 3: Lecture des fichiers INI et démarrage des démons](#)
  - [Etape 4: Communication de l'Arbiter vers les démons](#)
- [Exemple de connexions SSH en environnement multi-serveurs](#)

## Introduction

Shinken Entreprise comprend 7 démons ayant chacun une fonction.

- Suivant les principes Unix : un outil, une tâche, Shinken Entreprise a une architecture où chaque partie est isolée et se connecte aux autres avec une interface standard HTTP ou HTTPS.
- Basé sur un back-end HTTP ou HTTPS, cela vous permettra de construire une architecture distribuée et hautement disponible très simplement.

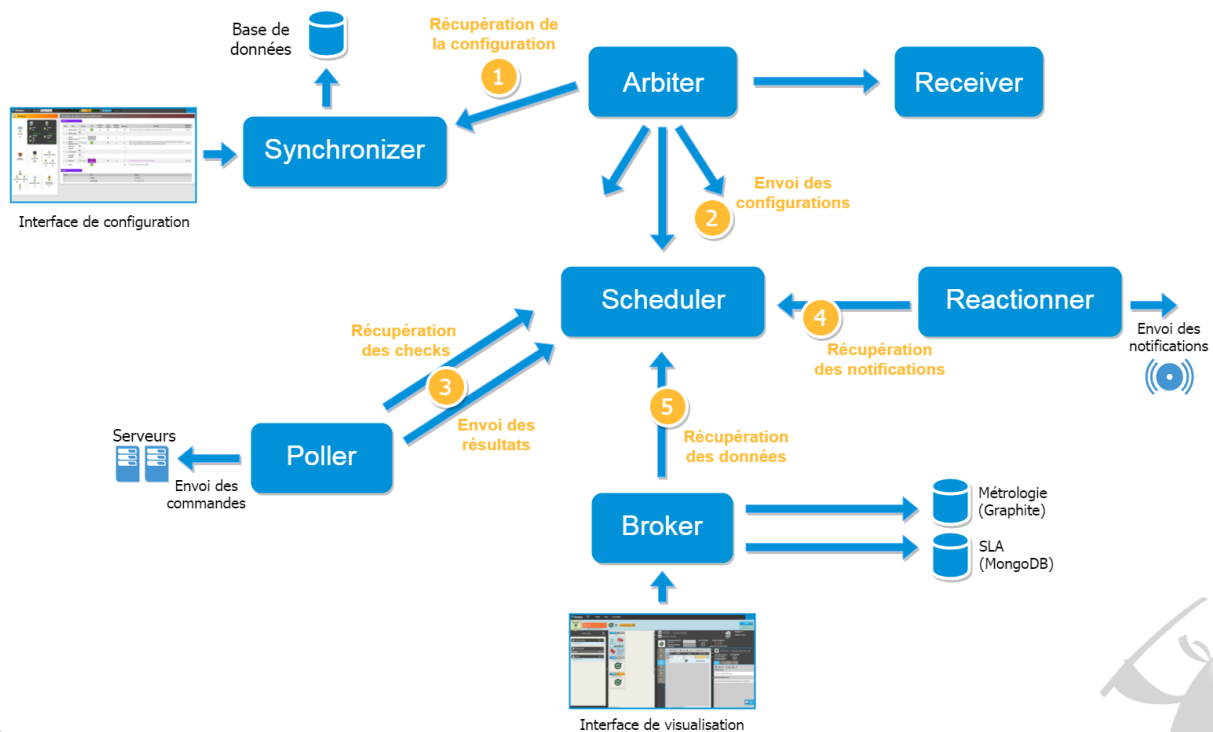
Il est possible de manipuler ces démons avec des commandes spéciales. Une page de documentation est dédiée à cet effet ici : [Outils en ligne de commande](#).

## Récapitulatif des différents démons

Ci-joint la table des différents démons, leur port par défaut, et leur rôle respectif :

Démon	Port d'écoute	Protocole	Rôle
<a href="#">Synchronizer</a>	7765	HTTP/HTTPS	Gère l'édition de la configuration
<a href="#">Synchronizer</a>	7777	HTTP/HTTPS	Gère le listener interne à Shinken
<a href="#">Arbiter</a>	7770	HTTP/HTTPS	Distribue la configuration à tous les démons
<a href="#">Poller</a>	7771	HTTP/HTTPS	Exécute les checks
<a href="#">Scheduler</a>	7768	HTTP/HTTPS	Analyse les statuts et contextes des éléments
<a href="#">Reactionner</a>	7769	HTTP/HTTPS	Envoie les notifications
<a href="#">Receiver</a>	7773	HTTP/HTTPS	Reçoit les résultats des checks externes ( <a href="#">Mode actif et mode passif</a> )
<a href="#">Broker</a>	7772	HTTP/HTTPS	Centralise et exporte les données

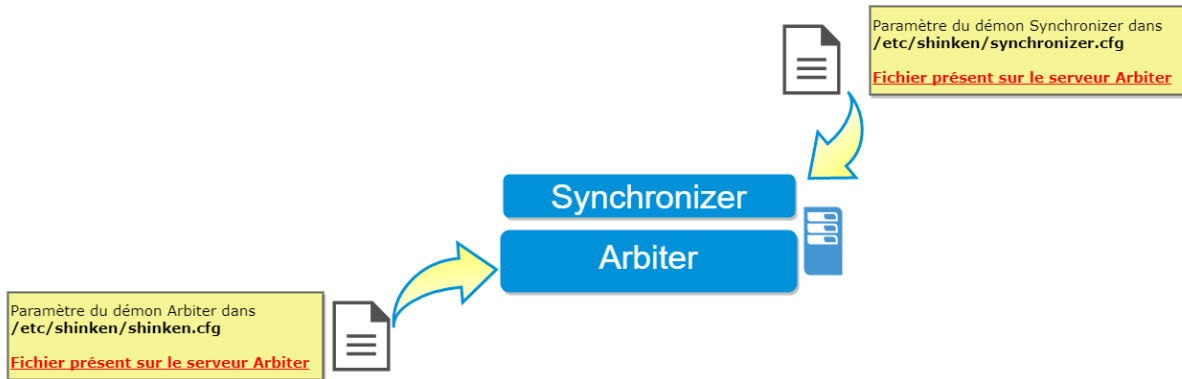
# Architecture Globale




Les étapes de démarrage des différents démons de l'installation Shinken

**Etape 1: Démarrage des démons Arbiter et Synchronizer**

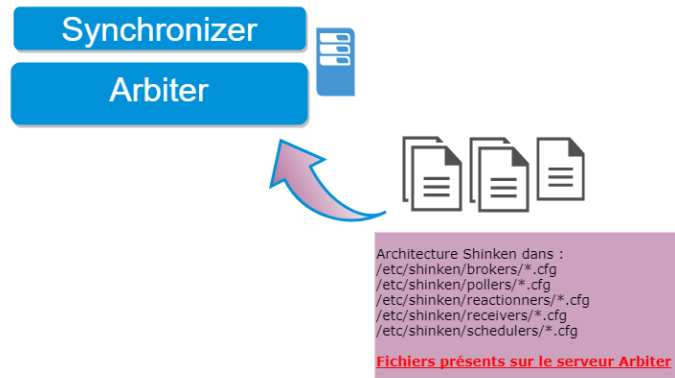
## Etape 1 : Démarrage des démons Arbiter et Synchronizer



 *Information : les paramètres du démon contiennent par exemple le port du socket utilisé pour la communication, le protocole utilisé (HTTP ou HTTPS), les certificats pour les SSL etc..*

## Etape 2: Lecture des fichiers CFG

## Etape 2 : Lecture des fichiers CFG de l'architecture Shinken par le démon Arbitre



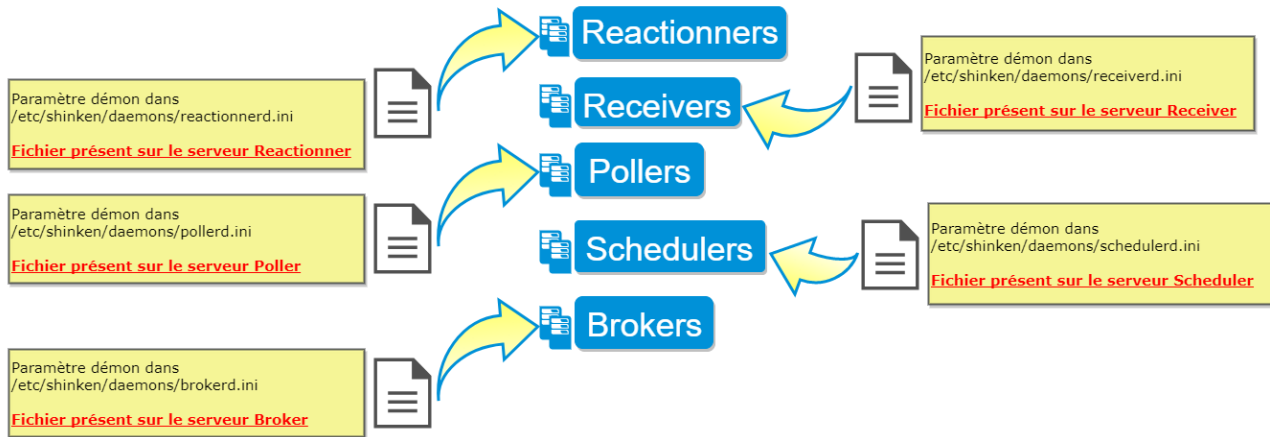
**⚓** Shinken n'a pas besoin des fichiers CFG des serveurs hébergeant les démons, il n'a besoin que des CFG du serveur hébergeant le démon Arbitre (central).

*Information : d'autres fichiers CFG sont également traités depuis le répertoire /etc/shinken/ de l'Arbiter, comme entres autres les CFG des répertoires "realms", "modules", "\_default",...*

*La liste exhaustive se trouve dans le fichier /etc/shinken/shinken.cfg*

## Etape 3: Lecture des fichiers INI et démarrage des démons

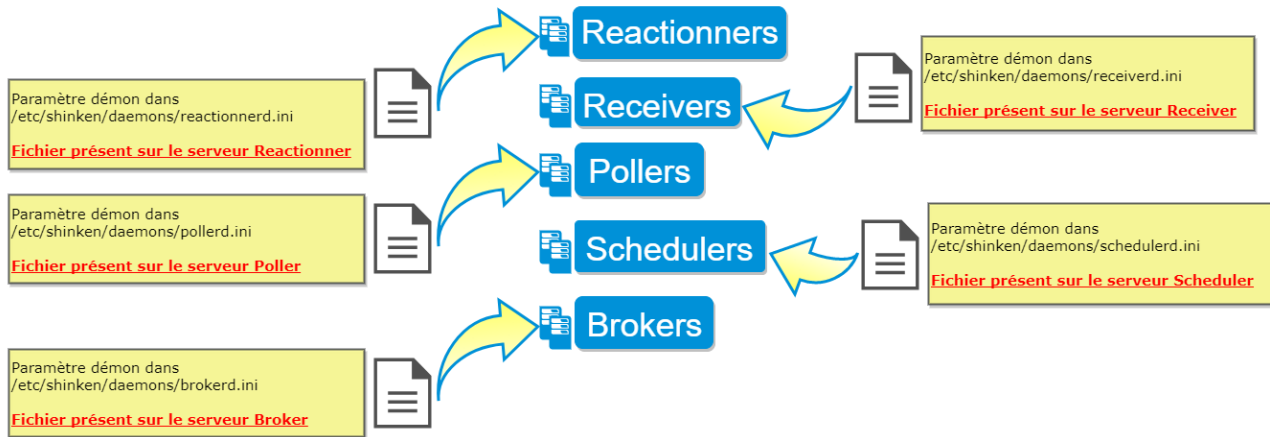
### Etape 3 : Lecture des fichiers INI et Démarrage des démons



**Information :** Dans une architecture distribuée, ces démons peuvent être placés sur des serveurs séparés et/ou distants. Chaque démon utilisera le fichier INI du serveur sur lequel il est hébergé. Les paramètres du démon contiennent par exemple le port du socket utilisé pour la communication, le protocole utilisé (HTTP ou HTTPS), les certificats pour les SSL etc..

### Etape 4: Communication de l'Arbiter vers les démons

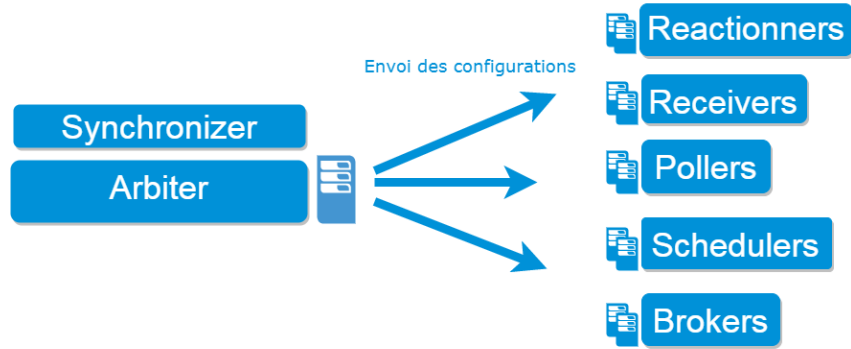
### Etape 3 : Lecture des fichiers INI et Démarrage des démons



**Information :** Dans une architecture distribuée, ces démons peuvent être placés sur des serveurs séparés et/ou distants  
Chaque démon utilisera le fichier INI du serveur sur lequel il est hébergé. Les paramètres du démon contiennent par exemple le port du socket utilisé pour la communication, le protocole utilisé (HTTP ou HTTPS), les certificats pour les SSL etc..

Exemple de connexions SSH en environnement multi-serveurs

## Etape 4 : Communication de l'Arbiter avec les démons



**Information :** L'Arbiter communique avec les démons via les ports de communication dédiés à cet effet. Il envoie les différentes configurations pour que les démons puissent réaliser leurs tâches respectives dans l'architecture.