

Realms

Shinken's architecture allows the administrator to have a unique point of administration with many schedulers, pollers, reactionners and brokers. Hosts are dispatched with their own services to schedulers and the satellites (pollers/reactionners/brokers) get jobs from them. Everyone is happy.

Or almost everyone. Think about an administrator who has a distributed architecture around the world. With the current Shinken Enterprise architecture the administrator can put a couple of scheduler/poller daemons in Europe and another set in Asia, but he cannot "tag" hosts in Asia to be checked by the asian scheduler. Also trying to check an asian server with an european scheduler can be very sub-optimal, read very sloooow. The hosts are dispatched to all schedulers and satellites so the administrator cannot be sure that asian hosts will be checked by the asian monitoring servers.

In the normal Shinken Enterprise Architecture is useful for load balancing with high availability, for single site.

Shinken Enterprise provides a way to manage different geographic or organizational sites.

We will use a generic term for this site management, **Realms**.

Realms in few words

A realm is a pool of resources (scheduler, poller, reactionner, receiver and broker) that hosts or hostgroups that can be attached to. A host or hostgroup can be attached to only one realm. All "dependancies" or parents of this hosts must be in the same realm. A realm can be tagged "default" and realm untagged hosts will be put into it. In a realm, pollers, reactionners and brokers will only get jobs from schedulers of the same realm.

Realms are different than the poller_tags

Make sure to understand when to use realms and when to use poller_tags.

- realms are used to segregate schedulers
- poller_tags are used to segregate pollers

For some cases poller_tag functionality could also be done using Realms. The question you need to ask yourself: Is a poller_tag "enough", or do you need to fully segregate a the scheduler level and use Realms. In realms, schedulers do not communicate with schedulers from other Realms:

- If you just need a poller in a DMZ network: use poller_tag
- If you need a scheduler/poller in a customer LAN: use realms

Sub realms

A realm can contain another realm. It does not change anything for schedulers: they are only responsible for hosts of their realm not the ones of the sub realms. The realm tree is useful for satellites like reactionners or brokers: they can get jobs from the schedulers of their realm, but also from schedulers of sub realms. Pollers can also get jobs from sub realms, but it's less useful so it's disabled by default. Warning: having more than one broker in a scheduler is not a good idea. The jobs for brokers can be taken by only one broker. For the Arbiter it does not change a thing: there is still only one Arbiter and one configuration whatever realms you have.

Example of realm usage

Let's take a look at two distributed environnements. In the first case the administrator wants totally distinct daemons. In the second one he just wants the schedulers/pollers to be distincts, but still have one place to send notifications (reactionners) and one place for database export (broker).

Distincts realms :

[blocked URL](#)

More common usage, the global realm with reactionner/broker, and sub realms with schedulers/pollers :

[blocked URL](#)

Satellites can be used for their realm or sub realms too. It's just a parameter in the configuration of the element.