

Implicit Roles

Authors	Neil Conway < neil@mesosphere.io > Adam Bordelon < adam@mesosphere.io >
Revision	0.3
Status	Final

Table of Contents

[Implicit Roles](#)

[Motivation](#)

[Proposed Behavior](#)

[Weights](#)

[HTTP Endpoints](#)

[Backward Compatibility](#)

[Comparison with Dynamic Roles](#)

[Future Work](#)

[Dynamic ACLs](#)

[Dynamic Weights](#)

Epic: [MESOS-3988](#)

Motivation

Mesos currently uses a static list of roles, specified when the master is started (via the `--roles` flag). A framework can only register as a role if (a) the role appears in the list of roles specified on master startup, and (b) the framework's principal is authorized to register as the role.

A static list of roles is inflexible: as cluster usage changes (e.g., new frameworks are installed), the set of roles will change over time. Requiring that all the masters be rebooted to change the allowed set of roles impairs usage of Mesos features that depend on roles, such as dynamic reservations, persistent volumes, and quota. Moreover, the list of roles is redundant with the ACLs: if an administrator wants to specify that some/all principals can only register as one of a fixed list of roles, that can be specified by setting the appropriate ACLs.

Proposed Behavior

The operator will no longer need to configure a list of roles: in a sense, *all possible roles* will be allowed by default (no ACLs). Hence, which role(s) a framework can register as will be determined only by authorization (ACLs). Similarly, which roles can make static and dynamic reservations will be determined only by authorization (ACLs).

In the short-term, the set of ACLs is static and must be configured on master startup (via the `--acls` flag). That will be addressed in a separate project (see below). In the meantime, implicit roles with static ACLs will still be useful: for example, by default Mesos does not enforce authorization (“ACLs.permissive” is true), so a framework will be allowed to register as any role. For low-security installations, this might be acceptable. Similarly, administrators can define broad ACL policies, such as specifying that the “trusted” principal can register with *ANY* role, while the “untrusted” principal can only register with the `limited-resources` role.

Weights

At present, a role can be assigned a *weight*, which influences resource allocation via DRF: a role with a weight of two is offered twice the fair-share of a role with a weight of one. Weights are configured statically (via the `--weights` flag) at master startup.

Switching to implicit roles will not change this functionality: in the short-term, weights will still be configured statically via the `--weights` flag. After introducing implicit roles, a weight can be assigned to *any* role name; roles that do not have a configured weight will be assumed to have a weight of one. Dynamic weights will be implemented as a separate project (see below).

If no frameworks are registered as a role, no resources will be removed for that role, so there should be no change in resource allocation behavior.

HTTP Endpoints

The `/roles` endpoint currently returns the (static) list of configured roles. For each role, we list the role’s weight, any resources reserved for that role, and any frameworks that have registered as that role.

With the introduction of implicit roles, this behavior will change slightly: if `--roles` is not specified, a role will appear in `/roles` iff it has a non-default weight, if it has any reserved resources, or if there are any frameworks currently registered in that role.

Backward Compatibility

To support backward compatibility, implicit roles will be introduced in two phases:

1. In the initial phase (goal: Mesos 0.27 release), “implicit roles” will only be enabled if the `--roles` command-line flag is **not** specified. If the `--roles` flag is specified, a warning will be printed to the log, and the old behavior (hard-coded list of roles) will be preserved.
2. After one deprecation cycle (~six months, Mesos ~0.33 release), an error will be raised if the `--roles` command-line flag is specified.

Comparison with Dynamic Roles

There is a concurrent proposal for dynamic roles ([MESOS-3177](#)). That design proposes that the set of roles be stored in the replicated log and that new operator endpoints be added to allow the set of roles to be modified.

Configuring the set of roles separately from the set of ACLs seems like additional complexity: with implicit roles, there is no need to store the roles in the replicated log or provide an additional set of endpoints for manipulating roles.

Future Work

Dynamic ACLs

Dynamic ACLs are clearly a requirement ([MESOS-2092](#)). However, there are situations in which implicit roles are useful even with the present, static ACL support (see examples above), so there seems to be value in implementing implicit roles first, as a separate project.

Dynamic Weights

Dynamic weights would be another useful (although lower priority) enhancement. This feature would involve:

- Storing weights in the replicated log
- Adding operator endpoints for changing the set of weights
- Deprecating the `--weights` command-line flag