Multihost Event Using Indexing and Iteration

Contents

Introduction

There are a number of ways to launch an event across multiple hosts. The most obvious are:

1. Create multiple events, one for each host.
2. Create a template, create multiple symlinks to the template.

In this example, we will show how to use a single event to do the same.

Single Event without Failover Support

The one caveat to this approach is that we cannot signal a problem using the `failover_event`. I.e., regardless of launch success or failure, the next event will be the same.

multihost

```
HOSTS=:mach1:mach2:mach3
HOST=$HOSTS[#HCRON_SELF_CHAIN]

as_user=
host=$HOST
command=/bin/true
notify_email=<emailaddr>
notify_message=<message>
when_month=*
when_day=*
when_hour=*
when_minute=*
when_dow=*
next_event=$HCRON_EVENT_NAME[-1]
failover_event=$HCRON_EVENT_NAME[-1]
```

Notes:

- `HOSTS` holds the list of machines to launch on.
- `HOSTS` splits as ["", "mach1", "mach2", "mach3"].
- `#HCRON_SELF_CHAIN` starts at 1, skipping the empty entry in `HOSTS`.
- Both `next_event` and `failover_event` call the event itself (`$HCRON_EVENT_NAME[-1]`).
- The "loop" ends when `#HCRON_SELF_CHAIN` is greater than the number of hosts in `HOSTS`.

Multi Event with Failover Support

If we need to be able to signal on a failure via the `failover_event`, we can use the following setup:

multihost

```
HOSTS=:mach1::mach2::mach3
HOST=$HOSTS[#HCRON_SELF_CHAIN]

as_user=
host=$HOST
command=/bin/true
notify_email=<emailaddr>
notify_message=<message>
when_month=*
when_day=*
when_hour=*
when_minute=*
when_dow=*
next_event=multihost-null
failover_event=multihost-fail
```

Notes:

- `HOSTS` splits as ["", "mach1", "", "mach2", "", "mach3"].
- `next_event` calls a "dummy" event which only serves to increase `$HCRON_SELF_CHAIN`.
- Regardless of whether the `next_event` or `failover_event` is called, the `$HCRON_SELF_CHAIN` gets updated and the `#HCRON_SELF_CHAIN` will be 1, 3, 5, ...

multihost-null

```
as_user=
host=localhost
command=/bin/true
notify_email=
notify_message=
when_year=2000
when_month=*
when_day=*
when_hour=*
when_minute=*
when_dow=*
next_event=$HCRON_EVENT_NAME[-1]
failover_event=$HCRON_EVENT_NAME[-1]
```

Notes:

- Serves as a "dummy" event.
- `when_year=2000` ensures it is never scheduled, even if it is not ignored.
- `$HCRON_SELF_CHAIN` updated with event name.
- Guarantees to call the original event no matter how it is called.

multihost-fail

```
as_user=
host=localhost
command=/bin/true
notify_email=<emailaddr>
notify_message=<message>
when_year=2000
when_month=*
when_day=*
when_hour=*
when_minute=*
when_dow=*
next_event=$HCRON_EVENT_NAME[-1]
failover_event=$HCRON_EVENT_NAME[-1]
```

Notes:

- Some command or email notification can be set up to send alert.
- `when_year=2000` ensures it is never scheduled, even if it is not ignored.
- `$HCRON_SELF_CHAIN` updated with event name.
- Guarantees to call the original event (`$HCRON_EVENT_NAME[-1]`) no matter how it is called.

Conclusion

In some situations, we may not want to create separate events for each host we want to launch on. These methods provide alternatives using event chaining and indexing. In effect, given a :-separated list of hosts (e.g., `HOSTS`) and specially configured `next_event` and `failover_events` settings, we can select a host to launch on using `#HCRON_SELF_CHAIN` as an index to iterate over the host list.