



## SCHEDULING

## Scheduling

### Custom Scheduler

If the default scheduling mechanisms (affinity, taints, policies) are not flexible enough for your needs, you can write your own scheduler. The programming of a custom scheduler is outside the scope of this course, but you may want to start with the existing scheduler code, which can be found in the [Scheduler repository on GitHub](#).

If a Pod specification does not declare which scheduler to use, the standard scheduler is used by default. If the Pod declares a scheduler, and that container is not running, the Pod would remain in a **Pending** state forever.

The end result of the scheduling process is that a pod gets a binding that specifies which node it should run on. A binding is a Kubernetes API primitive in the **api/v1** group. Technically, without any scheduler running, you could still schedule a pod on a node, by specifying a binding for that pod.

You can also run multiple schedulers simultaneously.

You can view the scheduler and other information with this command:

```
$ kubectl get events
```