Device Plugin Beta Graduation

According to the official documentation a beta feature is subject to the following requirements:

- *The version names contain beta (e.g. v2beta3).*
- ***Code is well tested**. **Enabling the feature is considered safe**. Enabled by default.*
- *Support for the overall feature will not be dropped, though details may change.*
- *The schema and/or semantics of objects may change in incompatible ways in a subsequent beta or stable release. When this happens, **we will provide instructions for migrating to the next version**. This may require deleting, editing, and re-creating API objects. The editing process may require some thought. This may require downtime for applications that rely on the feature.*
- *Recommended for only non-business-critical uses because of potential for incompatible changes in subsequent releases. If you have multiple clusters which can be upgraded independently, you may be able to relax this restriction.*

Moreover during the development cycles, some additional objectives were stated:

- Multiple device plugin implementations (not just GPU support)
    - This is to make sure that the device plugin API is a good fit for more than one device (i.e: not A GPU device plugin)
- Probe based registration model

Probe Based Registration Model

- Implementation is on github
- Design Document
- Introduces a new gRPC API
- Expected to be adopted by the CSI and the Device Manager
    - Do we have people from the CSI working on this?
    - Changes only affect the device plugin now
- Three way handshaking? Stream?
- Still some uncertainty about socket naming
- Still a WIP
    - No tests yet
    - Not yet reviewed by all the impacted groups:
        - Device Manager
        - CSI
        - sig-node
        - hack/OWNERS (should be fairly straightforward)

Multiple device plugin implementations

GKE GPU Device Plugin

- Are graphical images (Vulkan, OpenGL, …) supported?

NVIDIA GPU Device Plugin

- Need to setup the nvidia runtime as the default runtime
  - For dockershim (sadly this is the default runtime for most people)
  - For CRI-O in 1.10 annotations will allow to bypass setting the default runtime
- Graphical images need to be handled at runtime because:
  - We don't have access to pod / container information in the DPI
  - Some work needs to be done as a Pre-start hook
- GPU pods have an undefined behavior in case of restart or InitContainers
  - Being discussed here
  - Timeout is a blocker right now

FPGA support

- Kubelet device plugin APIs: Deallocate
  - Power Management
  - Security (memory scrubbing)
- Pass additional information about Pods to Allocate call
  - Providing right set of additional libraries, based on workload in the containers
  - Flexibility: Some devices might have fine-tune switches that can be better tuned by the workload
  - Development and debugging: annotations can trigger some development behavior inside device plugins

SolarFlare support

- Needs the pod name
  - Currently parses the kubelet checkpoint data to read that pod name
- Is now being mostly discussed in the Network Plumbing Working Group

KubeVirt VFIO support

- Plugin to expose VFIO devices
- Still WIP and some gaps to KubeVirt need to be closed

KubeVirt KVM support

- Requirement is to get /dev/kvm into a pod, in order to keep the container unprivileged
- Plugin to get /dev/kvm into a container

- Currently just a POC, but hopefully emerging into a long term solution
- Still WIP and some gaps to KubeVirt need to be closed

KubeVirt Network support

- Working POC to provide a network DP in order to provide L2 connectivity to a VM

General API state

- Display extended resources in node allocated resources
- [API BREAK] Change the multiple Allocate call to a single Pod Admit call
    - Related: Device plugin handling of partial allocation failures
- Support for resource quota on extended resources
- Device plugin can only advertise one resource type per registration.
- The design doc needs to be updated