Here, we bring together information regarding how to control your computing resources on the C3 AI Suite.

**Jupyter**

**Access Through Static Console**

The jupyter service is controlled through the 'Jupyter' type. To start the jupyter service with the default container profile, execute `Jupyter.start()` within the static console. To stop the jupyter service execute `Jupyter.stop()`.

To launch jupyter service with a specific config, use `Jupyter.startWithConfig()`. This accepts an argument of type `JupyterServiceConfig`. The `JupyterServiceConfig` can represent multiple types of services one of which is the 'resourceProfile'. This can represent pre-designated resource profiles of type `ResourceProfile`. To list the available resource profiles do `c3Grid(ResourceProfile.fetch())`.

By default, two `ResourceProfiles` exist, 'Basic', and 'BasicGPU'. For example, to launch the Jupyter service using the 'BasicGPU' profile, you would execute

```javascript
Jupyter.startWithConfig({"resourceProfile": "BasicGPU"})
```

**Creating new Resource Profile**

As a default, tags come with "Basic" and "BasicGPU" profiles. In case these do not fit your needs, it is possible to create a new profile from your static console using:

```javascript
var rp = ResourceProfile.upsert({'id': 'YourProfileName', 'name': 'YourProfileName', 'cpuCount': <cpus>,
    'memoryMb': <cpu_memory>, 'diskGb': <disk_memory>, 'gpuCount': <gpus>, 'gpuMemoryMb': <gpu_memory>});
```

If you want to make this your default profile whenever launching a Jupyter service, you can do so with:

```javascript
Jupyter.setDefaultResourceProfile('YourProfileName')
```

**Access Through IDS**

Through IDS, clear controls regarding Jupyter are exposed under the 'ML Studio' section for Application. When you've selected a project, you need to start the Jupyter service to access notebooks on that project. In the creation process, you're able to select the container profile to use for your jupyter service. By default, the 'BasicGPU' profile offers a single K80 GPU.

You can create new container profiles. Select the 'App Settings' menu, and there's a tab called 'Container Profiles'. You can create new profiles which have different resources like more GPUs, and more vCPUs, and more RAM. Create a new container profile, then when creating the Jupyter service select the new profile to gain access to those resources.