

**** DEA/DHA deployment prototype****

This is a continuation of the specific libvirt deployment prototype into a generic concept supporting a hardware plugin architecture in the deployment engine.

Conceptually the deployer contains of a number of entities:

- **The main deployment engine, `deploy.sh`. The deploy script needs three pieces of information:**
 - The ISO file to deploy
 - The `dea.yaml` file describing the Fuel deployment
 - The `dha.yaml` file describing the hardware configuration
- The Deployment Hardware Adapters (one per support hardware type). The adapter is an implementation of the DHA API for a specific hardware.
- The Deployment Hardware Adapter configuration (`dha.yaml`). The DHA configuration specifies the hardware configuration in terms of number of nodes and includes both general properties and specific information for the hardware adapter (such as IPMI configuration, libvirt VM names etc).
- The Deployment Environment Adapter configuration (`dea.yaml`). The DEA configuration describes an actual Fuel deployment, complete with network settings, node roles, interface configurations and more. The nodes identities in the `dea.yaml` must line up with those in the `dha.yaml`.

Both the `dea.yaml` and `dha.yaml` can be created from an existing Fuel deployment, in a way making a xerox copy of it for re-deployment. For this, the `create_templates` structure is copied to the Fuel master and the `create_templates.sh` is run there.

In the `examples/libvirt` directory, VM and network definitions for libvirt together with matching `dea.yaml` and `dha.yaml` can be found. The DEA configuration is made using a `opnfv-59` deployment.

There is also a hybrid libirt/IPMI adapter with an example `dea.yaml` and `dha.yaml` for a small one controller + one compute deploy in `examples/ipmi`.

The details and API description for DEA and DHA can be found in the documentation directory.

See the READMEs in the examples directories to get going with a Fuel deployment for your environment - or write and contribute your own hardware adapter for your environment!