OpenStorage API usage

Any storage product that uses the openstorage API can be managed via this API. Below are some examples of using this API.

Enumerate nodes in a cluster

```
import (
    ...

    "github.com/libopenstorage/gossip/types"
    "github.com/libopenstorage/openstorage/api"
    "github.com/libopenstorage/openstorage/api/client/cluster"
)

type myapp struct {
    manager cluster.Cluster
}

func (c *myapp) init() {
    // Choose the default version.
    // Leave the host blank to use the local UNIX socket, or pass in an IP and a port at wh
    clnt, err := cluster.NewClusterClient("", cluster.APIVersion)
    if err != nil {
        fmt.Printf("Failed to initialize client library: %v\n", err)
        os.Exit(1)
    }
    c.manager = cluster.ClusterManager(clnt)
}

func (c *myapp) listNodes() {
    cluster, err := c.manager.Enumerate()
    if err != nil {
        cmdError(context, fn, err)
        return
    }

    // cluster is now a hashmap of nodes... do something useful with it:
    for _, n := range cluster.Nodes {

     }
}
```

Inspect a volume in a cluster

```
import (
    ...

    "github.com/libopenstorage/openstorage/api"
    volumeclient "github.com/libopenstorage/openstorage/api/client/volume"
    "github.com/libopenstorage/openstorage/volume"
)

type myapp struct {
    volDriver volume.VolumeDriver
}

func (c *myapp) init() {
    // Choose the default version.
    // Leave the host blank to use the local UNIX socket, or pass in an IP and a port at wh
    clnt, err := volumeclient.NewDriverClient("", v.name, volume.APIVersion)
    if err != nil {
        fmt.Printf("Failed to initialize client library: %v\n", err)
        os.Exit(1)
    }
    v.volDriver = volumeclient.VolumeDriver(clnt)
}

func (c *myapp) inspect(id string) {
    stats, err := v.volDriver.Stats(id, true)
    if err != nil {
        return
    }

    // stats is an object that has various volume properties and statistics.
}
```