

Big Data Analytics and Docker

THE THRILLA IN MANILA



Speakers

Dean Hildebrand
IBM Research



Bill Owen
IBM

Nilesh Bhosale
IBM



Michael Hines
IBM Research

Qi Ming Teng
IBM Research



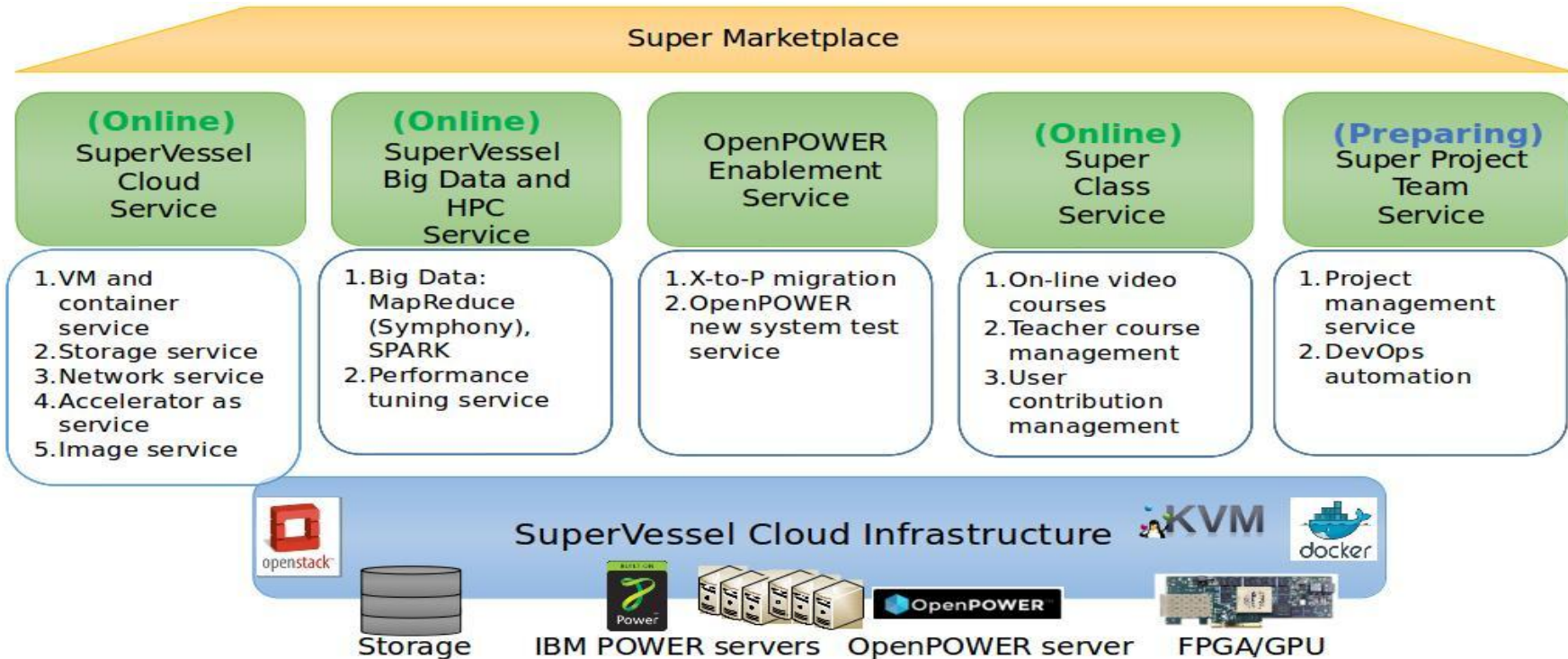
Outline

- What is the use case?
- What are the building blocks?
- How everything fits in together?
- Demo
- Challenges faced and Learnings



What is the use case?

- **SuperVessles Cloud** - University focussed cloud offering (www.ptopenlab.com)
- Cloud platform built using advanced technologies like OpenStack, Docker, Spark/Hadoop, Big Data Analytics, Cognitive Computing
- Open remote access to individual developers and university students
- Users from more than 30 universities from China, USA, Korea, Singapore, Africa



What are the building blocks?

- OpenStack Manila
- IBM Spectrum Scale (aka GPFS)
- Docker Containers
- OpenStack Heat
- Apache Spark

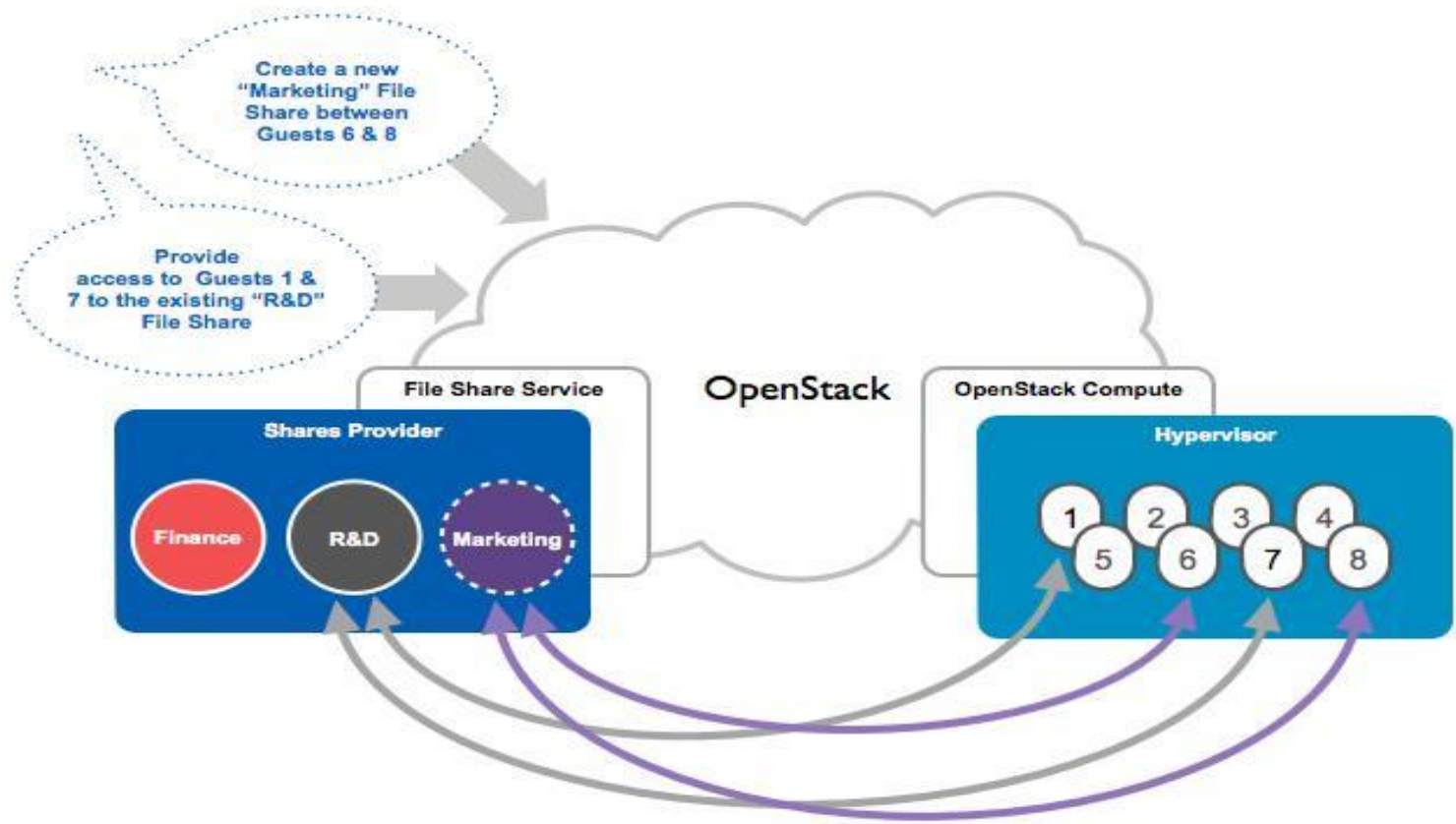


Manila: OpenStack Share File System Service

- Shared file system service across OpenStack Compute instances
- Vendor neutral APIs for provisioning and attaching filesystem-based storage such as NFS, CIFS, and other network file systems
- Share access control
- Multi-tenancy
- Operations:
 - Create/Delete/List file system shares
 - List, show, allow and deny access to file system shares
 - List share access rules
 - Create, list, and delete snapshots / clones of file systems shares

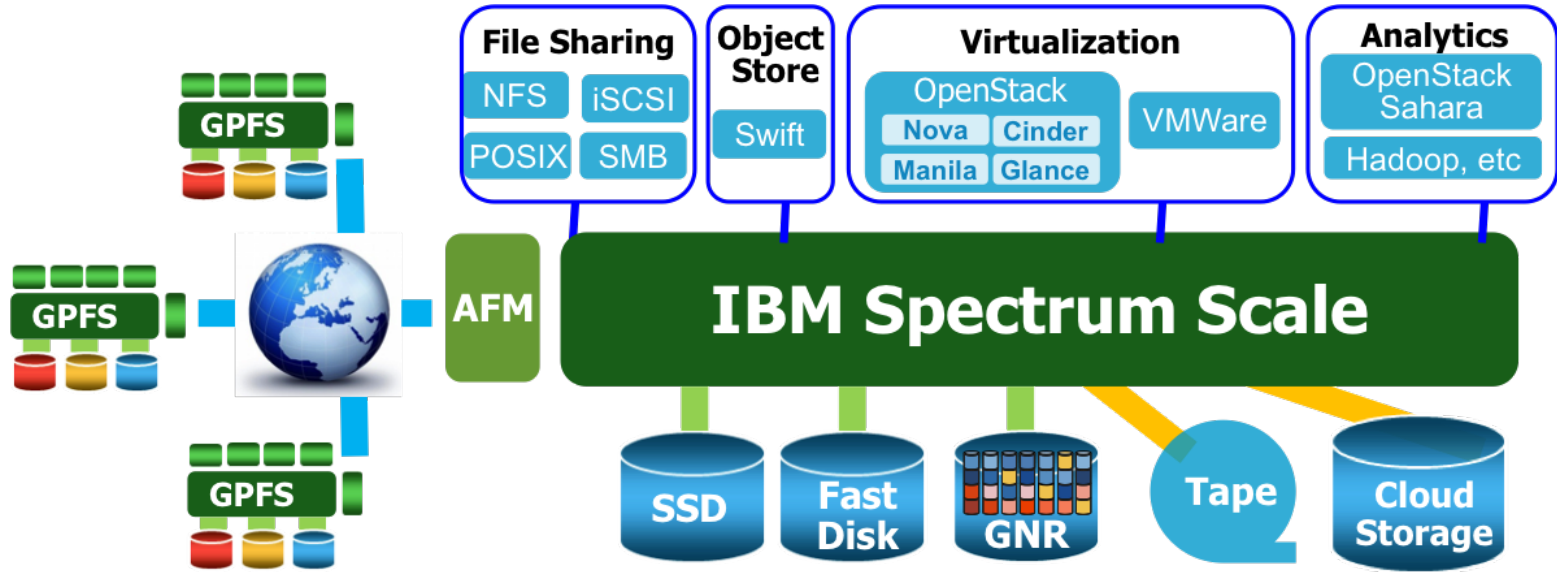


Manila: Typical Use Case



IBM Spectrum Scale Vision

Providing Single scale-out data plane for entire data center

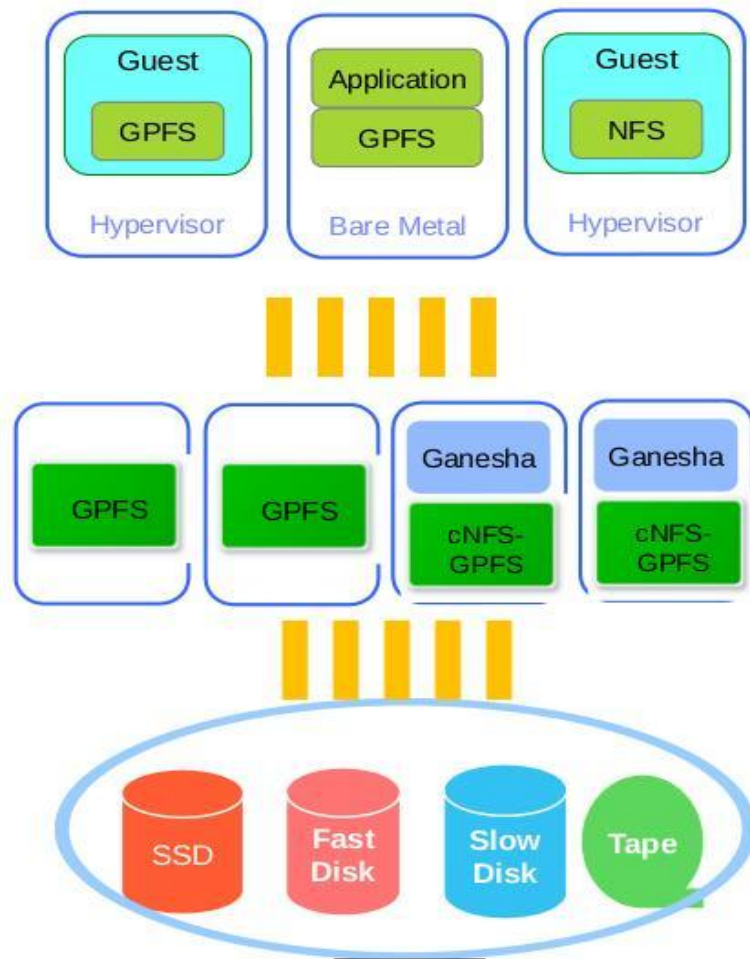


- **High Performance, Scale-out, Clustered File System**
- Unifies VM images, block devices, objects, and files
- Space Efficiency - GPFS Native RAID (GNR)
- Enterprise Features for Automatic Tiering, Data Distribution, Encryption, Migration to Tape and Cloud
- Data in **best location, on best tier, at the right time**

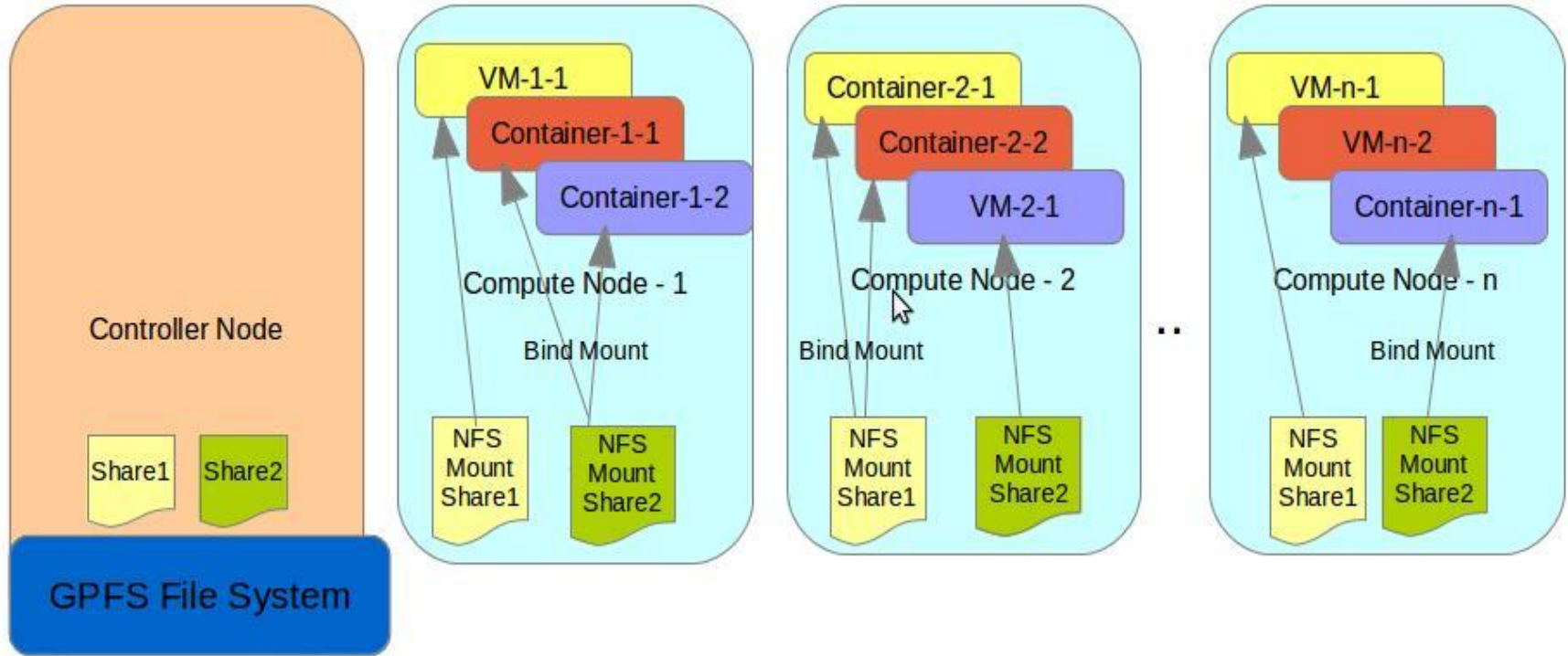


IBM Spectrum Scale w/ OpenStack Manila

- **Extends single data plane to VMs/Containers**
- Manila GPFS driver supports Kernel and Ganesha NFS
- Scale Out, High Performance, Highly Available, Encryption, Backup, DR, Declustered RAID
- Tiered Policy-based Storage Pools
- GPFS Fileset for isolation and quota enforcement for each tenant
- GPFS encryption provides further benefits such as secure delete
- GPFS Manila driver available with Kilo



Spectrum Scale - Manila Integration

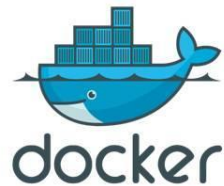


Controller Node: Running Manila services (m-api, m-sch, m-shr), NFS Server etc.

Compute Node: Running Nova-compute/Nova-Docker, Hypervisor, Nova network/Neutron



Docker Containers: Why and How?

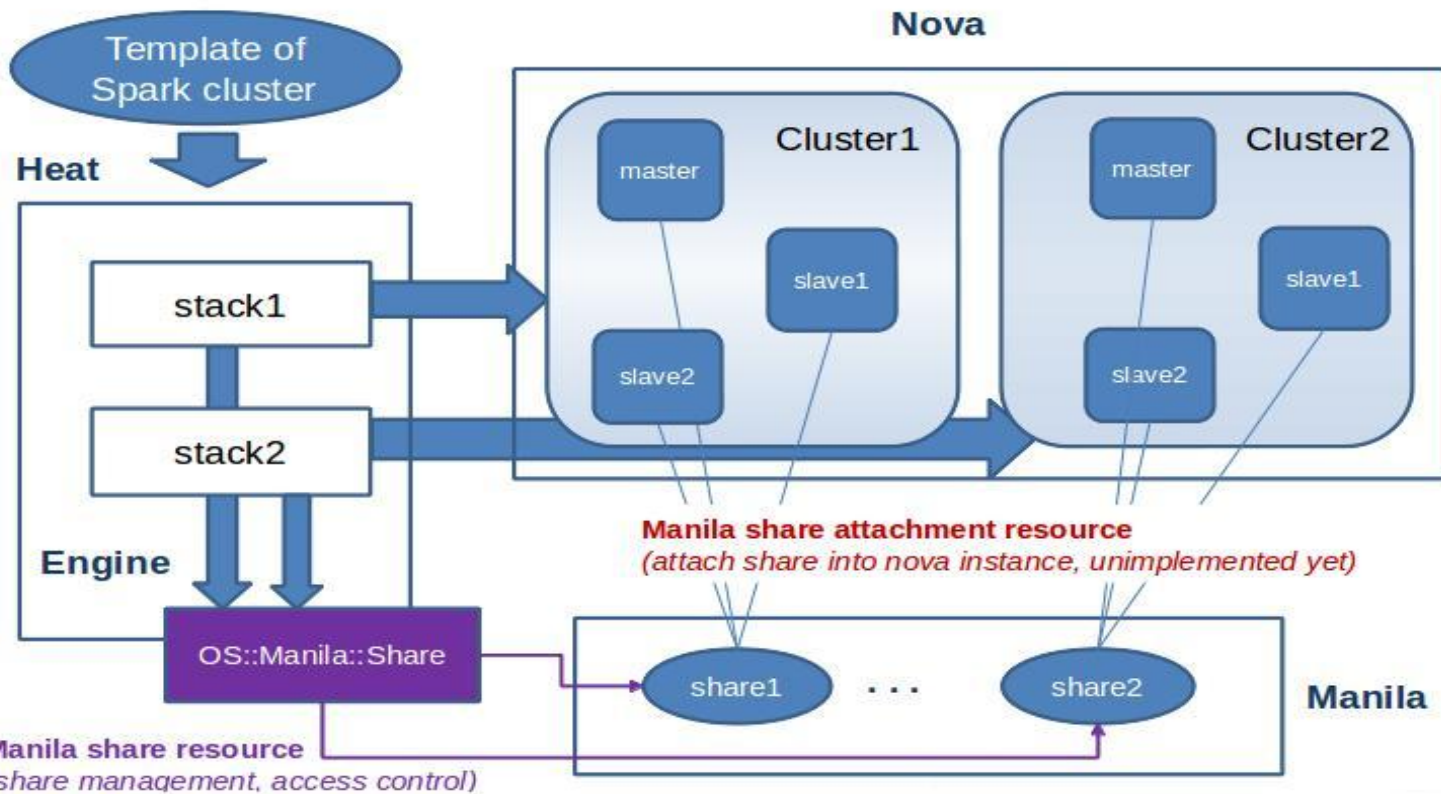


- **Lightweight** and **fast** as compared to VMs
- Used to run applications, OpenStack services, python daemons etc.
- Example Usage:
 - Deploying Manila services into openstack containers
 - Deploying big data services using HEAT into containers
- **Why Manila?**
 - Big Data processing requires shared file system
 - NFS with an OpenStack integrated service like Manila, providing the shared file system management, is a good option
 - Manila provides shared access to same data across containers
 - With HEAT, deployment can be automated in OpenStack environment



OpenStack HEAT: how do we use it with Manila?

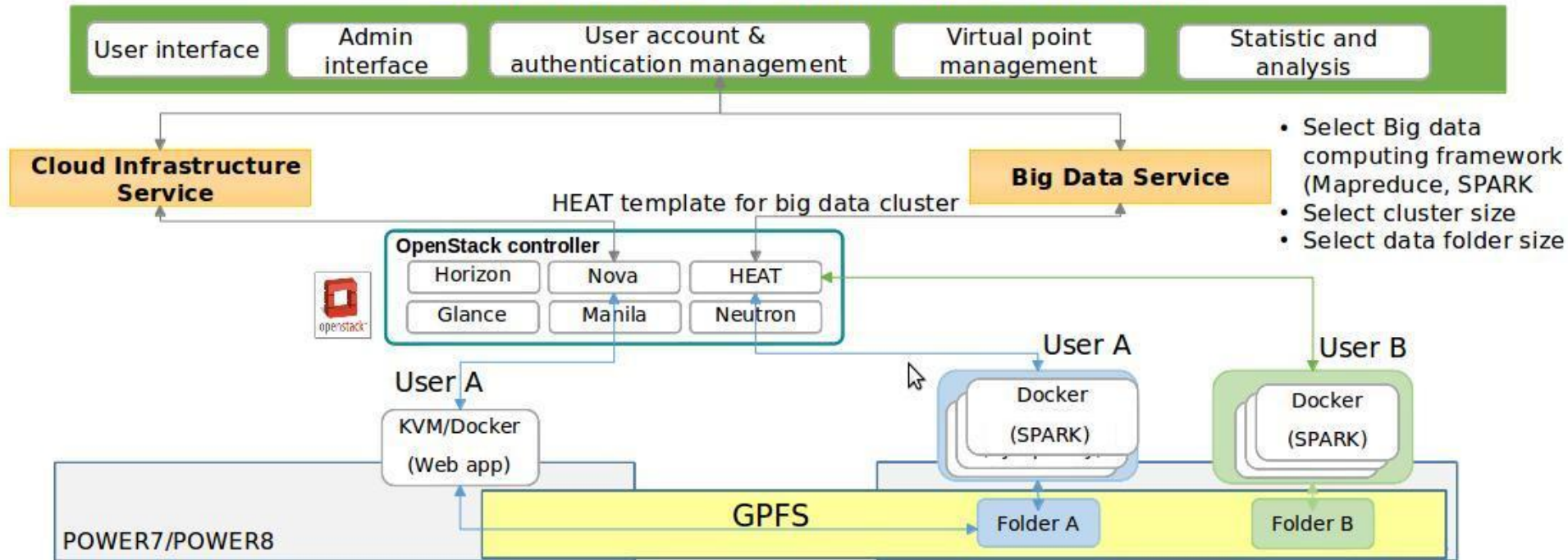
- An **Orchestration Engine** for OpenStack
- Based on templating mechanism
- Controls and orchestrates multiple composite cloud application



How everything fits in together?

OpenStack Manila, Docker, Heat, Spark, Hadoop: with SuperVessels Cloud

OpenPOWER-SuperVessel Unified Management Platform



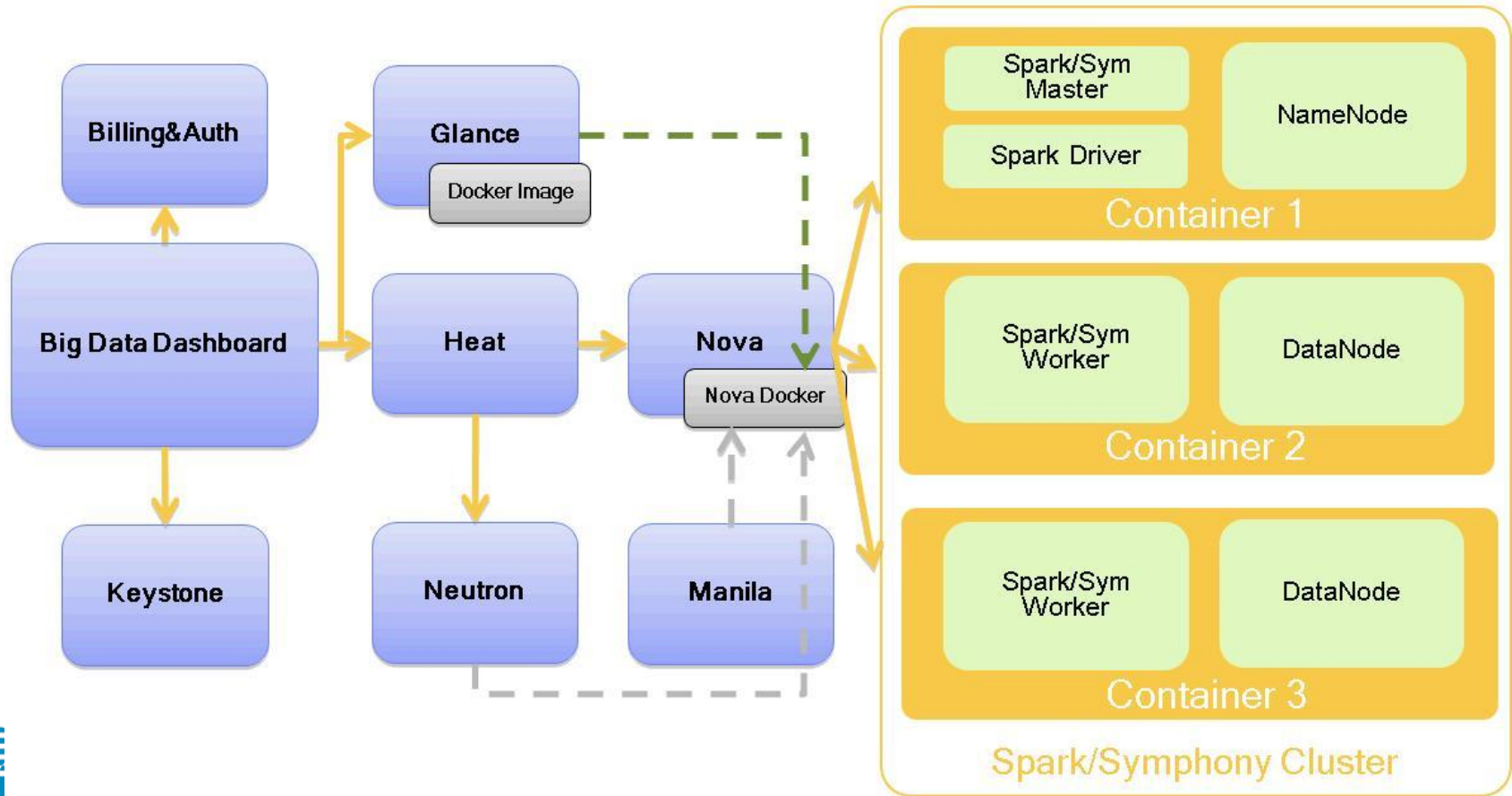
HEAT orchestrates docker instances, subnet and data folder based on user's request

Manila provides the NFS service using GPFS as backend, and the share is mounted via nova-docker

Share created by Manila could be accessed by the KVM/docker instances for big data analytics and other purpose



Deeper look ...



Demo



Challenges faced and Learnings

- Connecting storage to Docker is not currently available
We bind mount storage into the docker container
`$ docker run -v /var/lib/nova/{cinder|manila}/instance/mount_point /path/to/container`
(i.e. The python-equivalent of this, inside nova)
Mount the Manila share on host and then bind-mount inside the container
- Manila python-client in Juno does not support authentication based on auth-token
Applied patch from Kilo
- HEAT should not delete storage before docker has released it
Ordering for releasing resources in HEAT needs to be well designed
- Manila not yet supported in Ceilometer - no share specific metrics
Libcontainer doesn't export all the metrics either

