|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Hubs | Pod(is a group of one or more containers, with shared storage/network resources, and a specification for how to run the containers.) | API server | Registry (users pull images from registry instance) | Storage format | Builders(it is an instance,deployed on demand when it’s needed, on Google Cloud. | Graphical user portal system(:User can easily browse, search repositories and manage projects.) | full control of their environment |
| Docker |  | yes | yes | volumes |  | yes | no |
| Singularity |  | yes | yes | Yes( recipies) | yes | yes | yes |
| Kubernetes | yes | yes |  |  |  | yes |  |
| Quay |  | yes | yes |  |  | yes |  |
| Harbor |  | yes |  |  |  | yes |  |
| Portus |  | yes |  |  |  | yes |  |
| Docker Distribution |  | yes | yes( allows to configure registry) |  |  | yes |  |

Singularity Hub:

The Singularity software can import your Docker images without having Docker installed or being a superuser. Need to share your code? Put it in a Singularity container and your collaborator won’t have to go through the pain of installing missing dependencies. Do you need to run a different operating system entirely? You can “swap out” the operating system on your host for a different one within a Singularity container. As the user, you are in control of the extent to which your container interacts with its host. There can be seamless integration, or little to no communication at all.

//openshift

//software stack (system partition)  
2.Data Container: Created using sif tools that adds a data partition with the compressed information  
\*Unique identification of containers and repository (UUID+Github linked)  
Allow public and private repository settings  
Ensure security and confidentiality of containers

Team and user authentication

Read and write permissions

Grant version control of containers