|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Hubs | Store app and Data Container in SIF format | Preservation of immutable Containers | Identification | Public and private repository setting | Ensure security and confidentiality  of containers | Keep track of changes in container. |  |  |  |
| Docker | Docker uses the singularity build in command to build container in SIF format. | Yes(Docker images contain read-only layers, which means that once an image is created it is never modified.) | Volumes([https://docs.docker.com/storage/volumes/https://docs.docker.com/storage/volumes/](https://docs.docker.com/storage/volumes/)) | Yes( Docker provide an option to create public or private repositor  y when you build a repository) | Yes(<https://docs.docker.com/engine/security/>) | Layers(<https://medium.com/@jessgreb01/digging-into-docker-layers-c22f948ed612>)  Image IDs are 256 bits of random data and that’s pretty hard to guess |  |  |  |
| Kubernetes |  | Yes ( it provide the immutable container using pods. | A Kubernetes systems-generated string to uniquely identify objects.(<https://kubernetes.io/docs/concepts/overview/working-with-objects/names/>) | Yes ( At the beginning there will be an option to create private and public repository.) | Kubernetes (by default) assigns an IP address to every pod in the cluster and provides IP-based security(<http://techgenix.com/kubernetes-security-tools/>) |  |  |  |  |
| Singularity | Yes(compressed read-only **Singularity Image File (SIF)** format suitable for production (default) | Yes(**Singularity** production images are **immutable**. This is a feature added as of **Singularity** 2.4, and it ensures a higher level of reproducibility and verification of images. | Recipe file(<https://singularity.lbl.gov/docs-recipes>) | Yes(<https://researchit.las.iastate.edu/private-singularity-image-repositories>) | Yes(<https://singularity.lbl.gov/docs-security>) | Container Images(By default Singularity uses a container image file which can be checksummed, signed, and thus easily verified and/or validated.) |  |  |  |
| Quay |  | By adding Tag Immutability (TI) to quay you are able to do immutable container. | Each key is a unique ID for a container, with the value being a tuple of the type and configuration for that container.(<https://access.redhat.com/documentation/en-us/red_hat_quay/3/html/manage_red_hat_quay/quay-schema>) | The service is free for those who want to set up their own public repositories and available for a fee, if you want to create private repositories.(<https://access.redhat.com/articles/quayio-help>) | Repository mirroring and Quay Setup Operator.(<https://www.redhat.com/en/blog/red-hat-quay-32-welcome-container-security-operator>) | Tag the container to an image.(<https://docs.quay.io/solution/getting-started.html>) |  |  |  |
| Portus |  | yes  (Application tokens feature) | PORTUS\_PUMA\_TLS\_KEY and PORTUS\_PUMA\_TLS\_CERT | (<http://port.us.org/features/3_teams_namespaces_and_users.html>) | Yes(you can setup a security scanner for your Portus instance.) | Portus requires both a SSL key and a certificate to serve its contents over HTTPS. These files must be located in the /certificates directory of the container. Moreover, it's up to the deployer to set the PORTUS\_PUMA\_TLS\_KEY and PORTUS\_PUMA\_TLS\_CERT environment variables. Note that the key is also used to sign the JWT tokens issued to authenticate all the docker clients against the Registry. |  |  |  |
| Harbor |  | Create a Tag Immutability Rule(<https://goharbor.io/docs/1.10/working-with-projects/working-with-images/create-tag-immutability-rules/#create-a-tag-immutability-rule>) |  | Yes(<https://goharbor.io/docs/1.10/working-with-projects/create-projects/>) | Yes(Harbor has ‘deployment security’ which can prevent artifacts from being pulled if vulnerabilities are discovered) |  |  |  |  |
| OpenShift |  | Yes(<https://docs.openshift.com/container-platform/3.6/security/registries.html>) | OpenShift assigns a User ID (UID) range, a supplemental group ID (GID) range, and unique SELinux MCS labels to the project or namespace.(<https://www.openshift.com/blog/a-guide-to-openshift-and-uids>) | You can create a private repository. | Yes. OpenShift enables a secure software supply chain which ensures that applications are secure, without reducing developer productivity.(<https://www.openshift.com/learn/topics/security/>) |  |  |  |  |
| Docker Distribution |  |  | This repository's main product is Docker Registry implementation for storing and distributing Docker and OCI images using the OCI Distribution Specification. | It did not allow to make public and private repositories but it allow to make docker registry for storing and maintaining containers. |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |