
RDO container registry Documentation

Release 0.0.1.dev28

Red Hat

Jun 08, 2018

1	Table of Contents	3
1.1	About the registry	3
1.2	Installing the registry	3
1.2.1	Installing SSL certificates	3
1.2.2	Installing OpenShift Standalone Registry	4
1.3	How to get the service account tokens:	5
1.3.1	More reading	5
1.4	Using the registry	5
1.4.1	Pulling container images	5
1.4.2	Pushing container images	5
1.4.3	Tagging container images	6
1.4.4	Using the OpenShift Origin client	6
1.4.5	Container image metadata	7
1.5	Troubleshooting the registry	18



1.1 About the registry

RDO uses an [OpenShift standalone registry](#) which is more or less the upstream for the [Atomic Registry](#) project.

We chose to use the OpenShift standalone registry because it provides features that `docker-registry` and `docker-distribution` do not have out of the box.

Some reasons and features include but are not limited to:

- Being able to list images in the registry: `oc get imagestreams`
- Provide a web interface to browse and manage images in the registry
- Built-in authentication and access control (ACL) with GitHub oauth support
- Dogfood the OpenShift standalone registry use case and establish a feedback loop with OpenShift developers

1.2 Installing the registry

Warning: Fleshing out this documentation is a work in progress.

1.2.1 Installing SSL certificates

Certificates are provided by LetsEncrypt and are expected to have been generated prior to the installation. The certificate and key file paths are expected to have a specific format so that OpenShift doesn't overwrite existing files when copying them into its directories:

```
yum -y install git

# Retrieve letsencrypt and run it
git clone https://github.com/letsencrypt/letsencrypt
```

Now you can either:

- production: generate the SSL certificates with `./letsencrypt.sh`
- development: generate a CA and some dummy certificates with `./mock-certs.sh`

1.2.2 Installing OpenShift Standalone Registry

To install OpenShift Standalone Registry on localhost as root:

Set up local key-based authentication:

```
# We'll be connecting on localhost over ssh, setup keypair authentication
ssh-keygen -f ~/.ssh/id_rsa -t rsa -N ''
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
ssh-keyscan -H registry.rdo-project.org >> ~/.ssh/known_hosts
```

Install dependencies:

```
yum install -y python-setuptools python-devel libffi-devel openssl-devel redhat-rpm-
↪config git gcc
easy_install pip
pip install tox
```

Export oauth application credentials for github authentication:

```
export RDO_GITHUB_CLIENT_ID=oauth_client_id
export RDO_GITHUB_CLIENT_SECRET=oauth_client_secret
```

Note: `/var/lib/docker` will be set up on a separate block device with `docker-storage-setup`. If you do not provide the `host_preparation_docker_disk` variable for the host-preparation playbook, a loopback device will be generated with test purposes and the playbook will warn about it.

Note: The server stores an OpenShift persistent volume for the Docker registry on the local filesystem in `/openshift_volumes`. If you expect a high volume of data, you should re-mount this directory on a large partition or volume prior to installation.

Note: `ansible_ssh_user` **MUST** be provided for the `openshift-ansible` playbook, it is required by tasks such as `openshift_master_certificates` : Lookup default group for `ansible_ssh_user`.

Retrieve and run `rdo-container-registry` and `openshift-ansible` playbooks:

```
git clone https://github.com/rdo-infra/rdo-container-registry
cd rdo-container-registry
tox -e ansible-playbook -- -i hosts -e "host_preparation_docker_disk=/dev/vdb" host-
↪preparation.yml
# Note: https://github.com/openshift/openshift-ansible/issues/5812
# Glean configures "NM_CONTROLLED=no" in the ifcfg-eth0 file
tox -e ansible-playbook -- -i hosts openshift-ansible/playbooks/byo/openshift-node/
↪network_manager.yml -e "ansible_ssh_user=${USER}"
tox -e ansible-playbook -- -i hosts openshift-ansible/playbooks/byo/config.yml -e
↪"ansible_ssh_user=${USER}"
```

1.3 How to get the service account tokens:

Note: These operation is done directly on the master

Retrieve service account token for image pushes (for CI and things like that):

```
oc describe serviceaccount tripleo.service -n tripleo
oc describe secret tripleo.service-token-<generated> -n tripleo
```

1.3.1 More reading

- https://docs.openshift.com/container-platform/latest/admin_guide/manage_authorization_policy.html
- https://docs.openshift.com/container-platform/latest/dev_guide/projects.html
- https://docs.openshift.com/container-platform/latest/admin_guide/service_accounts.html

1.4 Using the registry

Warning: Fleshing out this documentation is a work in progress.

1.4.1 Pulling container images

```
docker pull trunk.registry.rdoproject.org/<namespace>/<container>(:<tag>)
```

Examples:

```
docker pull trunk.registry.rdoproject.org/tripleo/centos-binary-base:latest-passed-ci
docker pull trunk.registry.rdoproject.org/developer/centos
```

1.4.2 Pushing container images

Before you can push containers, you need to log in to the registry using the `docker login` command.

In order to use the `docker login` command, you need to obtain a token first. This token is obtained by logging on to the [registry console](#).

In order to log in to the registry console, you need to be a member of a specific [GitHub team](#) which grants the required access.

After you have successfully logged in to the console, the home page will provide the login command that you can copy & paste, it looks like this:

```
docker login -p abcdef_token -e unused -u unused trunk.registry.rdoproject.org
```

Afterwards, you may push container images to projects in which you have the necessary privileges, for example:

```
docker pull docker.io/centos
docker tag docker.io/centos trunk.registry.rdoproject.org/myproject/centos
docker push trunk.registry.rdoproject.org/myproject/centos
```

1.4.3 Tagging container images

With docker

When tagging images directly with Docker, keep in mind that your image names must contain the registry you'll be pushing to as well as the namespace or project:

```
docker pull docker.io/centos:latest
docker tag docker.io/centos:latest trunk.registry.rdoproject.org/tripleo/
↪centos:anothertag
docker push trunk.registry.rdoproject.org/tripleo/centos:anothertag
```

With the OpenShift Origin client

If the image is already in the registry and you just want to tag the image with a different tag, you can do it directly with the OpenShift Origin client without pulling and pushing the image:

```
oc tag tripleo/centos-binary-base:latest tripleo/centos-binary-base:anothertag
```

Tip: If you aren't familiar with the OpenShift Origin client, the next section explains how you can get started.

1.4.4 Using the OpenShift Origin client

The OpenShift Origin client allows you to query the registry to list images, get image metadata information or do operations on images.

To install the OpenShift client:

```
# On Fedora
dnf -y install origin-clients

# On CentOS
yum -y install centos-release-openshift-origin
yum -y install origin-clients
```

If you have an account

If you have an account and are able to log in to the [registry console](#), it will provide a login command that you can copy & paste, it looks like this:

```
oc login --token abcdef_token registry.rdoproject.org:8443
```

You will then be able to use `oc` commands against the projects you have access to.

If you do not have an account

If you do not have an account and you are not able to log in to the [registry console](#), you will only be able to list container images for projects that are *public* (*anonymous*).

You will need to create a configuration file in order to tell your OpenShift client where the OpenShift cluster is located in order to send it's queries.

This configuration file is located at `~/.kube/config` and needs to look like the following:

```
apiVersion: v1
clusters:
- cluster:
    server: https://registry.rdoproject.org:8443
    name: registry-rdoproject-org:8443
contexts:
- context:
    cluster: registry-rdoproject-org:8443
    namespace: default
kind: Config
preferences: {}
```

You will then be able to use `oc` commands against projects that are made public or anonymous.

Listing container images

The OpenShift client has the ability to list available container images in a project over CLI:

```
oc get imagestreams -n tripleo
NAME                                DOCKER REPO
↪ TAGS      UPDATED
centos-binary-aodh-api             172.30.132.198:5000/tripleo/centos-binary-aodh-api
↪ latest    2 hours ago
centos-binary-aodh-base            172.30.132.198:5000/tripleo/centos-binary-aodh-base
↪ latest    2 hours ago
centos-binary-aodh-evaluator        172.30.132.198:5000/tripleo/centos-binary-aodh-
↪ evaluator latest    2 hours ago
```

Note: Note that the DOCKER REPO field contains an internal URL. This will be improved to show the public registry endpoint in a future version of OpenShift, in the meantime, you can substitute that URL by `trunk.registry.rdoproject.org`.

1.4.5 Container image metadata

With the OpenShift client

`oc describe imagestreams` or `oc describe is`:

```
oc describe imagestreams centos-binary-aodh-api -n tripleo
Name: centos-binary-aodh-api
Namespace: tripleo
Created: 23 hours ago
Labels: <none>
Annotations: <none>
Docker Pull Spec: 172.30.132.198:5000/tripleo/centos-binary-aodh-api
Unique Images: 4
Tags: 4

latest
  pushed image
```

(continues on next page)

(continued from previous page)

```

* 172.30.132.198:5000/tripleo/centos-binary-aodh-
↪api@sha256:b558c7e942d03dbaf506cae0b8bba81ec98c4d132f8f81fdbdead1521ca6fd73
  23 hours ago

38471d4ccf3914805fafaa56b21db2cc83755e95_5d6d179f
  pushed image

* 172.30.132.198:5000/tripleo/centos-binary-aodh-
↪api@sha256:85efabdbdb663802d6387623fc9f76c13ad89f74bf121e8f246f0f9b22cd261e
  22 hours ago

87a9e523723c0707a56341e9a7f7542bb4ec9567_c928cd3f
  pushed image

* 172.30.132.198:5000/tripleo/centos-binary-aodh-
↪api@sha256:64c2837a84c7a72acfc2c633426cbb600b0771dde1259de7203a61a1b4c37aae
  23 hours ago

latest-passed-ci
  pushed image

* 172.30.132.198:5000/tripleo/centos-binary-aodh-
↪api@sha256:aa784a9b213b8d8b42b1ed09cbc3e6111956703cbd223f7f6f77f17c48383665
  22 hours ago

```

oc describe imagestreamtags or oc describe istags:

```

oc describe imagestreamtags centos-binary-aodh-api:latest -n tripleo
Name:          sha256:b558c7e942d03dbaf506cae0b8bba81ec98c4d132f8f81fdbdead1521ca6fd73
Namespace:     <none>
Created:       23 hours ago
Labels:        <none>
Annotations:   openshift.io/image.managed=true
Docker Image:  172.30.132.198:5000/tripleo/centos-binary-aodh-
↪api@sha256:b558c7e942d03dbaf506cae0b8bba81ec98c4d132f8f81fdbdead1521ca6fd73
Image Name:    sha256:b558c7e942d03dbaf506cae0b8bba81ec98c4d132f8f81fdbdead1521ca6fd73
Image Size:    237.1 MB (first layer 72.29 MB, last binary layer 545 B)
Image Created: 24 hours ago
Author:        <none>
Arch:          amd64
Command:       kolla_start
Working Dir:   <none>
User:          <none>
Exposes Ports: <none>
Docker Labels:
  build-date=20170529
  build_id=1496093093
  kolla_version=4.0.0
  license=GPLv2
  maintainer=TripleO Project (http://tripleo.org)
  name=aodh-api
  rdo_version=87a9e523723c0707a56341e9a7f7542bb4ec9567_c928cd3f
  vendor=CentOS
Environment:
  PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
  KOLLA_BASE_DISTRO=centos
  KOLLA_INSTALL_TYPE=binary

```

(continues on next page)

(continued from previous page)

```
KOLLA_INSTALL_METATYPE=rdo
PS1=$(tput bold) $(printenv KOLLA_SERVICE_NAME) $(tput sgr0) [$(id -un)@$(hostname_
↪-s) $(pwd)]$
```

With Skopeo

Skopeo is a tool from [Project Atomic](#) that allows you interact easily with Docker registries.

For example, it allows you to inspect metadata remotely without downloading the image first:

```
skopeo inspect docker://trunk.registry.rdoproject.org/tripleo/centos-binary-aodh-api
{
  "Name": "trunk.registry.rdoproject.org/tripleo/centos-binary-aodh-api",
  "Tag": "latest",
  "Digest": "sha256:b558c7e942d03dbaf506cae0b8bba81ec98c4d132f8f81fdbdead1521ca6fd73
↪",
  "RepoTags": [
    "latest",
    "38471d4ccf3914805fafaa56b21db2cc83755e95_5d6d179f",
    "87a9e523723c0707a56341e9a7f7542bb4ec9567_c928cd3f",
    "latest-passed-ci"
  ],
  "Created": "2017-05-29T21:29:21.715464885Z",
  "DockerVersion": "1.12.6",
  "Labels": {
    "build-date": "20170529",
    "build_id": "1496093093",
    "kolla_version": "4.0.0",
    "license": "GPLv2",
    "maintainer": "TripleO Project (http://tripleo.org)",
    "name": "aodh-api",
    "rdo_version": "87a9e523723c0707a56341e9a7f7542bb4ec9567_c928cd3f",
    "vendor": "CentOS"
  },
  "Architecture": "amd64",
  "Os": "linux",
  "Layers": [
    "sha256:dd6405a9d6445ac370348c852c1d28dbdbbbefca4a40f5a302d02ea59488023d",
    "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
    "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
    "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
    "sha256:2cd7b6008767ea2a93ac3d0ab8b034a008f74e4ad6cbefcdade1132ed0cd9357e",
    "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
    "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
    "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
    "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
    "sha256:0f9e56455f34aceba55c30e37223ecc9c78a699665e666949fe46f4522553c2",
    "sha256:2e83a0b6361f9d4a4ff6d34622f856ef853d5fb5d69380a2ce725da479a3a1bd",
    "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
    "sha256:bf6f0cf0995e7369a7cb3fc4de97a1252a9a33cce553971b68be89798f54a3c0",
    "sha256:72be68d311b8dfd0328e46ef8561fc4e8a44e3aa12d47e87332c1bdfb581398a",
    "sha256:bd2e425390db662ed28bc21372c7fa0256d130e55f7bf8ec6339fc41fa1d166c",
    "sha256:7020914f6824dda8830b3dbb559500e41cfb6de2c0c85b94ce8e42d874cad2b4",
    "sha256:c4f089101f2cebb3f9b57259fbc73137a1a1181f9eedc408f41b1392912e7c82",
    "sha256:33abc8109d965f93c302c9811928b6ebdf9bb22339e2cbdd8dda32f0e6f64461",
```

(continues on next page)

(continued from previous page)

```

        "sha256:bddef0ebe7c602b3ea948d0d9fe0707d4ecb02e78e33de77af279e18829cc280",
        "sha256:e83b73b2bea3e03313d3243240e7e30fd2ac585afbb57223053e1b658439b46c",
        "sha256:a3ece904123ddd878bdbd88aa01817de01b79f481df9965b9a5cd4f56e529b72",
        "sha256:54c91f9119b0a5e1925e3eb4c35df00268d99f76eace85b248c64b30ac896d1a",
        "sha256:702759ec043e579a7a5634a6fc1010ecf567af91069928571e472e7a69cbf4fb",
        "sha256:ea2620fc7ad861d9cf3f42cf6a4d73be4b25889a8a2610e854aefc508f5b5d0e",
        "sha256:8d8ec19a2094876862b81a2194bbe72efcab2ad7b6f2a59f91d75056b8eef64d",
        "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
        "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
        "sha256:da5c65ac8909f45a23d180f074a49d9b768d697d1db90dd79c813182c265ea80",
        "sha256:1b403b81203da688a9add9c41291a87f8c8f08cac4244540b9e5f66d1db30a06",
        "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
        "sha256:68ebdcb67a8ebe7f7d525dbbfd85e989181c965b52441fc6ff1c1b1a41477f9e",
        "sha256:b4a7f16092b091deff942a0a5cb5da27b63859d7af5422630f32a47f87f54bbf",
        "sha256:b892f4c6c3b7e139dbe8d7e403c4558f5b424c5cf07a51ff5ab074c6201cd779",
        "sha256:0fd273e790a9b2fc17d28ec6255094f5fe63a1576e54c208134c0435f1c1b2cc",
        "sha256:73786d94d5dedd650ce3193adfa2b92502131e90287731ac6d26db5de1ae5d4d",
        "sha256:fdcf22601cf27f222030e783f8755e11a1adb0dc949a7e735740e67c02548171",
        "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
        "sha256:3f59f43f810c97c60d62f42c8779ad9aadeecfb1647feb86630dcc444ab710995",
        "sha256:bd7ad955318a78798da9d8b4c9ad4c683bf38dcd93db7b2f6a1c111e37b563ff",
        "sha256:79743704eaad90ed9ce3a67fa1185babc3ffc9f015213831b659e67ed11704d5"
    ]
}

```

With curl

Note: The following examples are querying the OpenShift API directly, not the docker-registry server. For more information about OpenShift API endpoints and how you can use them, you can refer to the official OpenShift [documentation](#).

For imagestreams:

```

curl https://registry.rdoproject.org:8443/oapi/v1/namespaces/tripleo/imagestreams/
↪centos-binary-aodh-api
{
  "kind": "ImageStream",
  "apiVersion": "v1",
  "metadata": {
    "name": "centos-binary-aodh-api",
    "namespace": "tripleo",
    "selfLink": "/oapi/v1/namespaces/tripleo/imagestreams/centos-binary-aodh-api",
    "uid": "a0c69ef8-44c0-11e7-858e-fa163e033b7c",
    "resourceVersion": "3600",
    "generation": 1,
    "creationTimestamp": "2017-05-29T22:46:17Z"
  },
  "spec": {},
  "status": {
    "dockerImageRepository": "172.30.132.198:5000/tripleo/centos-binary-aodh-api",
    "tags": [
      {
        "tag": "latest",

```

(continues on next page)

(continued from previous page)

```

    "items": [
      {
        "created": "2017-05-29T22:46:17Z",
        "dockerImageReference": "172.30.132.198:5000/tripleo/centos-binary-aodh-
↪api@sha256:b558c7e942d03dbaf506cae0b8bba81ec98c4d132f8f81fdbdead1521ca6fd73",
        "image":
↪"sha256:b558c7e942d03dbaf506cae0b8bba81ec98c4d132f8f81fdbdead1521ca6fd73",
        "generation": 1
      }
    ]
  },
  {
    "tag": "38471d4ccf3914805fafaa56b21db2cc83755e95_5d6d179f",
    "items": [
      {
        "created": "2017-05-29T23:11:38Z",
        "dockerImageReference": "172.30.132.198:5000/tripleo/centos-binary-aodh-
↪api@sha256:85efabdbdb663802d6387623fc9f76c13ad89f74bf121e8f246f0f9b22cd261e",
        "image":
↪"sha256:85efabdbdb663802d6387623fc9f76c13ad89f74bf121e8f246f0f9b22cd261e",
        "generation": 1
      }
    ]
  },
  {
    "tag": "87a9e523723c0707a56341e9a7f7542bb4ec9567_c928cd3f",
    "items": [
      {
        "created": "2017-05-29T22:46:31Z",
        "dockerImageReference": "172.30.132.198:5000/tripleo/centos-binary-aodh-
↪api@sha256:64c2837a84c7a72acfc2c633426cbb600b0771dde1259de7203a61a1b4c37aae",
        "image":
↪"sha256:64c2837a84c7a72acfc2c633426cbb600b0771dde1259de7203a61a1b4c37aae",
        "generation": 1
      }
    ]
  },
  {
    "tag": "latest-passed-ci",
    "items": [
      {
        "created": "2017-05-29T23:11:24Z",
        "dockerImageReference": "172.30.132.198:5000/tripleo/centos-binary-aodh-
↪api@sha256:aa784a9b213b8d8b42bled09cbc3e6111956703cbd223f7f6f77f17c48383665",
        "image":
↪"sha256:aa784a9b213b8d8b42bled09cbc3e6111956703cbd223f7f6f77f17c48383665",
        "generation": 1
      }
    ]
  }
]
}
}

```

For `imagestreamtags`:

```

curl https://registry.rdoproject.org:8443/oapi/v1/namespaces/tripleo/imagestreamtags/
↪centos-binary-aodh-api:latest
{
  "kind": "ImageStreamTag",
  "apiVersion": "v1",
  "metadata": {
    "name": "centos-binary-aodh-api:latest",
    "namespace": "tripleo",
    "selfLink": "/oapi/v1/namespaces/tripleo/imagestreamtags/centos-binary-aodh-api
↪%3Alatest",
    "uid": "a0c69ef8-44c0-11e7-858e-fa163e033b7c",
    "resourceVersion": "3600",
    "creationTimestamp": "2017-05-29T22:46:17Z"
  },
  "tag": null,
  "generation": 1,
  "image": {
    "metadata": {
      "name": "sha256:b558c7e942d03dbaf506cae0b8bba81ec98c4d132f8f81fdbdead1521ca6fd73
↪",
      "uid": "a0c7a764-44c0-11e7-858e-fa163e033b7c",
      "resourceVersion": "3098",
      "creationTimestamp": "2017-05-29T22:46:17Z",
      "annotations": {
        "openshift.io/image.managed": "true"
      }
    },
    "dockerImageReference": "172.30.132.198:5000/tripleo/centos-binary-aodh-
↪api@sha256:b558c7e942d03dbaf506cae0b8bba81ec98c4d132f8f81fdbdead1521ca6fd73",
    "dockerImageMetadata": {
      "kind": "DockerImage",
      "apiVersion": "1.0",
      "Id": "11e27e00f85f340a8ad1e9e2330dc61b80f3ea962db9c468c7776a21f6ceee00",
      "Parent": "ad9dfe0c7ccc21ff769cb644a30f88f8ddad536fec46664a23537cf91cad33e",
      "Created": "2017-05-29T21:29:21Z",
      "Container": "f019f8ffc684ac4a9ed476d2ef53cf4612e7a4889ae7bc6e0c32d1743b524753",
      "ContainerConfig": {
        "Hostname": "dfa0e46aa7ac",
        "Env": [
          "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
          "KOLLA_BASE_DISTRO=centos",
          "KOLLA_INSTALL_TYPE=binary",
          "KOLLA_INSTALL_METATYPE=rdo",
          "PS1=$(tput bold) $(printenv KOLLA_SERVICE_NAME) $(tput sgr0) ${id -un}@
↪$(hostname -s) $(pwd)]$ "
        ],
        "Cmd": [
          "/bin/sh",
          "-c",
          "chmod 755 /usr/local/bin/kolla_aodh_extend_start"
        ],
        "Image":
↪"sha256:b2c4694ba4d018bbfe55f5546091684bbe51e14f18b5aa53aaaff24d81b07c61",
        "Labels": {
          "build-date": "20170529",
          "build_id": "1496093093",
          "kolla_version": "4.0.0",

```

(continues on next page)

(continued from previous page)

```

        "license": "GPLv2",
        "maintainer": "TripleO Project (http://tripleo.org)",
        "name": "aodh-api",
        "rdo_version": "87a9e523723c0707a56341e9a7f7542bb4ec9567_c928cd3f",
        "vendor": "CentOS"
    },
    },
    "DockerVersion": "1.12.6",
    "Config": {
        "Hostname": "dfa0e46aa7ac",
        "Env": [
            "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
            "KOLLA_BASE_DISTRO=centos",
            "KOLLA_INSTALL_TYPE=binary",
            "KOLLA_INSTALL_METATYPE=rdo",
            "PS1=$(tput bold) $(printenv KOLLA_SERVICE_NAME) $(tput sgr0) [$(id -un)@
↪$(hostname -s) $(pwd)]$ "
        ],
        "Cmd": [
            "kolla_start"
        ],
        "Image":
↪"sha256:b2c4694ba4d018bbfe55f5546091684bbe51e14f18b5aa53aaaff24d81b07c61",
        "Labels": {
            "build-date": "20170529",
            "build_id": "1496093093",
            "kolla_version": "4.0.0",
            "license": "GPLv2",
            "maintainer": "TripleO Project (http://tripleo.org)",
            "name": "aodh-api",
            "rdo_version": "87a9e523723c0707a56341e9a7f7542bb4ec9567_c928cd3f",
            "vendor": "CentOS"
        }
    },
    "Architecture": "amd64",
    "Size": 237061207
},
"dockerImageMetadataVersion": "1.0",
"dockerImageLayers": [
    {
        "name":
↪"sha256:dd6405a9d6445ac370348c852c1d28dbdbbbefca4a40f5a302d02ea59488023d",
        "size": 72292482,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪"sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
        "size": 32,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪"sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
        "size": 32,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    }
],

```

(continues on next page)

(continued from previous page)

```

    {
      "name":
↪ "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
      "size": 32,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:2cd7b6008767ea2a93ac3d0ab8b034a008f74e4ad6cbefcdade1132ed0cd9357e",
      "size": 22717,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
      "size": 32,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
      "size": 32,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
      "size": 32,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:0f9e56455f34aceaba55c30e37223ecc9c78a699665e666949fe46f4522553c2",
      "size": 266,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:2e83a0b6361f9d4a4ff6d34622f856ef853d5fb5d69380a2ce725da479a3a1bd",
      "size": 528,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
      "size": 32,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:bf6f0cf0995e7369a7cb3fc4de97a1252a9a33cce553971b68be89798f54a3c0",

```

(continues on next page)

(continued from previous page)

```

        "size": 2023,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:72be68d311b8dfd0328e46ef8561fc4e8a44e3aa12d47e87332c1bdfb581398a",
        "size": 1598,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:bd2e425390db662ed28bc21372c7fa0256d130e55f7bf8ec6339fc41fa1d166c",
        "size": 1621,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:7020914f6824dda8830b3dbb559500e41cfb6de2c0c85b94ce8e42d874cad2b4",
        "size": 5360567,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:c4f089101f2cebb3f9b57259fbe73137a1a1181f9eedc408f41b1392912e7c82",
        "size": 5485035,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:33abc8109d965f93c302c9811928b6ebdf9bb22339e2cbdd8dda32f0e6f64461",
        "size": 5370612,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:bddef0ebe7c602b3ea948d0d9fe0707d4ecb02e78e33de77af279e18829cc280",
        "size": 34553499,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:e83b73b2bea3e03313d3243240e7e30fd2ac585afbb57223053e1b658439b46c",
        "size": 3923,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:a3ece904123ddd878bdbd88aa01817de01b79f481df9965b9a5cd4f56e529b72",
        "size": 596,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:54c91f9119b0a5e1925e3eb4c35df00268d99f76eace85b248c64b30ac896d1a",
        "size": 546,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    },

```

(continues on next page)

(continued from previous page)

```

    {
      "name":
↪ "sha256:702759ec043e579a7a5634a6fc1010ecf567af91069928571e472e7a69cbf4fb",
      "size": 250,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:ea2620fc7ad861d9cf3f42cf6a4d73be4b25889a8a2610e854aefc508f5b5d0e",
      "size": 27474,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:8d8ec19a2094876862b81a2194bbe72efcab2ad7b6f2a59f91d75056b8eef64d",
      "size": 4838,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
      "size": 32,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
      "size": 32,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:da5c65ac8909f45a23d180f074a49d9b768d697d1db90dd79c813182c265ea80",
      "size": 14847530,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:1b403b81203da688a9add9c41291a87f8c8f08cac4244540b9e5f66d1db30a06",
      "size": 54587991,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
      "size": 32,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:68ebdcb67a8ebe7f7d525dbbfd85e989181c965b52441fc6ff1c1b1a41477f9e",
      "size": 2785,
      "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
      "name":
↪ "sha256:b4a7f16092b091deff942a0a5cb5da27b63859d7af5422630f32a47f87f54bbf",

```

(continues on next page)

(continued from previous page)

```

        "size": 37298747,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:b892f4c6c3b7e139dbe8d7e403c4558f5b424c5cf07a51ff5ab074c6201cd779",
        "size": 8871,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:0fd273e790a9b2fc17d28ec6255094f5fe63a1576e54c208134c0435f1c1b2cc",
        "size": 359,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:73786d94d5dedd650ce3193adfa2b92502131e90287731ac6d26db5de1ae5d4d",
        "size": 239,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:fdcf22601cf27f222030e783f8755e11a1adb0dc949a7e735740e67c02548171",
        "size": 565,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4",
        "size": 32,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:3f59f43f810c97c60d62f42c8779ad9aadeafb1647feb86630dcc444ab710995",
        "size": 7184422,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:bd7ad955318a78798da9d8b4c9ad4c683bf38dcd93db7b2f6a1c111e37b563ff",
        "size": 546,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    },
    {
        "name":
↪ "sha256:79743704eaad90ed9ce3a67fa1185bab33ffc9f015213831b659e67ed11704d5",
        "size": 545,
        "mediaType": "application/vnd.docker.container.image.rootfs.diff+x-gtar"
    }
    ],
    "dockerImageSignatures": [
↪ "eyJJoZWFKZXIiOnsia2RlIjpw7ImNydiI6IlAtMjU2Iiwia2lkIjoim1dXTzpWVlpGOjRUR0I6TzJPQTpSV083OkxINDQ6TkZPU"
↪ "
    ],

```

(continues on next page)

(continued from previous page)

```

    "dockerImageManifestMediaType": "application/vnd.docker.distribution.manifest.
    ↪v1+json"
  }
}

```

1.5 Troubleshooting the registry

Warning: Fleshing out this documentation is a work in progress.

```

# Logs for the OpenShift processes
journalctl -u origin-master-api --follow
journalctl -u origin-master-controllers --follow
journalctl -u origin-node --follow

# Note, commands using -n default is to select from the default namespace

# List routes, pods, services and deployment configurations
oc get routes -n default
oc get pods -n default
oc get svc -n default
oc get dc -n default

# Dump configuration of things
oc export routes -n default -o yaml |less
oc export pods -n default -o yaml |less
oc export svc -n default -o yaml |less
oc export dc -n default -o yaml |less

# Follow logs from running pods
oc get pods -n default
oc logs -f -n default <pod name> (ex: oc logs -f -n default docker-registry-1-xgxb)

# Execute a command in a running pod
oc get pods -n default
oc exec -n default <pod name> <command> (ex: oc exec -n default docker-registry-1-
    ↪xgxb ls)

# Get a shell on a running pod
oc get pods -n default
oc rsh <pod name> -n default (ex: oc rsh docker-registry-1-xgxb -n default)

# Look at policies and permissions for a project
oc get rolebindings -n project

# If authentication on the master node doesn't seem right
# You might be logged on as a different user
oc whoami
oc login -u system:admin --config=/etc/origin/master/admin.kubeconfig
oadm config get-contexts
oadm config use-context default/192-168-1-17:8443/system:admin

```