**Overview of Kubernetes Upgrade**

StarlingX Kubernetes upgrade enables you to upgrade the Kubernetes components on a running system from one supported version to another supported version.

StarlingX only supports specific Kubernetes versions and upgrades paths. For each supported Kubernetes version, it’s tracked via metadata in the Sysinv component. The patching mechanism is used to deliver metadata and software in platform patches to enable upgrades for specific Kubernetes versions. Typically, three structured patches must be uploaded/applied/installed properly before starting the upgrades.

* Patch1
* platform patch for sysinv and playbookconfig components
* contains metadata for new Kubernetes version
* contains the templates of k8s networking pods for the new Kubernetes version
* Patch2
* Kubernetes patch that patches kubeadm RPM to the new Kubernetes version
* Patch3
* Kubernetes patch that patches kubelet and kubectl RPMs to the new kubernetes version

Here is a high level summary of the kubernetes upgrade process:

* The upgrades process starts by downloading Kubernetes images for the new Kubernetes version.
* After the images are downloaded, the control plane upgrades can be triggered on either of the controllers to upgrade kube-apiserver, kube-controller-manager and kube-scheduler pods.

Before upgrading the second control plane on the other controller, the Kubernetes networking upgrades that upgrade networking pods must be completed first. Then kubelet upgrades must be done first on controller nodes, then worker nodes. Except for All-in-one simplex system, before upgrading each kubelet, the host must be locked. Host-unlock is required after the kubelet is upgraded on the host . The kubelet can be upgraded in parallel on worker nodes as long as there is sufficient capacity remaining on other worker hosts. No upgrades actions needed for storage hosts as they are not in the Kubernetes cluster.

* After the kubelet upgrades are all completed, the final step of the upgrade process is to complete the upgrade.

If any of the upgrade steps fails, the step can be re-done to continue upgrading.

**Upgrading Kubernetes Cluster**

**Pre-requisites**

* The system is clear of alarms
* all hosts must be unlocked, enabled and available
* Kubernetes pods must be all ready
* The installed applications are compatible with the new Kubernetes version
* All patches required for the new Kubernetes version must be transferred to the active controller.

**Procedure**

1. Upload/apply/install the platform patch.

The sw-patch commands (upload, apply, host-install) should be used to install the patch. No additional host-lock/unlock steps are required.

1. List available kubernetes versions.

$ system kube-version-list

+---------+--------+-----------+

| Version | Target | State |

+---------+--------+-----------+

| v1.16.1 | True | active |

| v1.16.2 | False | available |

+---------+--------+-----------+

Note:

This list comes from metadata in the sysinv component.

The fields:

* Target: denotes currently selected for installation.
* State:
* active: version is running everywhere
* partial: version is running somewhere
* available: version that can be upgraded to

1. Upload/apply/install kubeadm patch.

The sw-patch commands (upload, apply, host-install) should be used to install the patch. No additional host-lock/unlock steps are required.

1. Upload kubelet patch.

The sw-patch upload <kubelet-patch> should be used. Do not apply kubelet patch as it cannot be applied before start upgrading kubelet.

1. Start kubernetes upgrade.

$ system kube-upgrade-start v1.16.2

+-------------------+-------------------+

| Property | Value |

+-------------------+-------------------+

| from\_version | v1.16.1 |

| to\_version | v1.16.2 |

| state | upgrade-started |

+-------------------+-------------------+

As part of the upgrade start, the upgrade process checks the applied/available patches, the upgrade path, the health of the system, the installed applications compatibility and validates the system is ready for an upgrade.

**NOTE**: Use the command system kube-upgrade-start --force to force the upgrades process to

start and to ignore management affecting alarms. This should ONLY be done if you feel these alarms will not be an issue over the upgrades process.

The states include:

upgrade-started: semantic checks passed, upgrade started

downloading-images: images downloading in progress

downloaded-images: images downloading complete

downloading-images-failed: images downloading fail

upgrading-first-master: first master node control plane upgrade in progress

upgraded-first-master: first master node control plane upgrade complete

upgrading-first-master-failed: first master node control plane upgrade fail

upgrading-networking: networking plugin upgrade in progress

upgraded-networking: networking plugin upgrade complete

upgrading-networking-failed: networking plugin upgrade fail

upgrading-second-master: second master node control plane upgrade in progress

upgraded-second-master: second master node control plane upgrade complete

upgrading-second-master-failed: second master node control plane upgrade fail

upgrading-kubelets: kubelet upgrades in progress

upgrade-complete: all nodes upgraded

1. Download kubernetes images.

$ system kube-upgrade-download-images

+--------------+--------------------------------------+

| Property | Value |

+--------------+--------------------------------------+

| uuid | b5f7dada-2537-4416-9d2c-f9ca9fcd0e22 |

| from\_version | v1.16.1 |

| to\_version | v1.16.2 |

| state | downloading-images |

| created\_at | 2020-02-20T16:08:48.854869+00:00 |

| updated\_at | None |

+--------------+--------------------------------------+

The state “downloaded-images” entered when kubernetes images download is complete.

To verify the action is completed, use system kube-upgrade-show to check the upgrade state.

$ system kube-upgrade-show

+--------------+--------------------------------------+

| Property | Value |

+--------------+--------------------------------------+

| uuid | b5f7dada-2537-4416-9d2c-f9ca9fcd0e22 |

| from\_version | v1.16.1 |

| to\_version | v1.16.2 |

| state | downloaded-images |

| created\_at | 2020-02-20T16:08:48.854869+00:00 |

| updated\_at | 2020-02-20T16:10:37.858661+00:00 |

+--------------+--------------------------------------+

1. Upgrade control plane on first controller.

$ system kube-host-upgrade controller-1 control-plane

+-----------------------+-------------------------+

| Property | Value |

+-----------------------+-------------------------+

| control\_plane\_version | v1.16.1 |

| hostname | controller-1 |

| id | 2 |

| kubelet\_version | v1.16.1 |

| personality | controller |

| status | upgrading-control-plane |

| target\_version | v1.16.2 |

+-----------------------+-------------------------+

Either controller can be upgraded first.

The state “upgraded-first-master” entered when the first control plane upgrade is done.

1. Upgrade kubernetes networking.

$ system kube-upgrade-networking

+--------------+--------------------------------------+

| Property | Value |

+--------------+--------------------------------------+

| uuid | b5f7dada-2537-4416-9d2c-f9ca9fcd0e22 |

| from\_version | v1.16.1 |

| to\_version | v1.16.2 |

| state | upgrading-networking |

| created\_at | 2020-02-20T16:08:48.854869+00:00 |

| updated\_at | 2020-02-20T16:18:11.459736+00:00 |

+--------------+--------------------------------------+

Networking upgrade must be done after the first control plane is upgraded and before start upgrading the second control plane.

The state “upgraded-networking” entered when the networking upgrade is done.

1. Upgrade control plane on second controller

$ system kube-host-upgrade controller-0 control-plane

+-----------------------+-------------------------+

| Property | Value |

+-----------------------+-------------------------+

| control\_plane\_version | v1.16.1 |

| hostname | controller-0 |

| id | 1 |

| kubelet\_version | v1.16.1 |

| personality | controller |

| status | upgrading-control-plane |

| target\_version | v1.16.2 |

+-----------------------+-------------------------+

The state “upgraded-second-master” entered when the second control plane upgrade is done.

1. Show kubernetes upgrade status for hosts

$ system kube-host-upgrade-list

+----+--------------+-------------+----------------+-----------------------+-----------------+--------+

| id | hostname | personality | target\_version | control\_plane\_version | kubelet\_version | status |

+----+--------------+-------------+----------------+-----------------------+-----------------+--------+

| 1 | controller-0 | controller | v1.16.2 | v1.16.2 | v1.16.1 | None |

| 2 | controller-1 | controller | v1.16.2 | v1.16.2 | v1.16.1 | None |

| 3 | storage-0 | storage | v1.16.1 | N/A | N/A | None |

| 4 | storage-1 | storage | v1.16.1 | N/A | N/A | None |

| 5 | worker-0 | worker | v1.16.1 | N/A | v1.16.1 | None |

| 6 | worker- 1 | worker | v1.16.1 | N/A | v1.16.1 | None |

+----+--------------+-------------+----------------+-----------------------+-----------------+--------+

The control planes of both controllers are upgraded to v1.16.2 now.

1. Apply/Install kubelet/kubectl patch

The sw-patch commands (apply, host-install) should be used to install the patch. This will place the new version of kubelet binary on each host, but will not restart kubelet.

1. Upgrade kubelet on each controller

$ system host-lock controller-1

$ system kube-host-upgrade controller-1 kubelet

+-----------------------+-------------------+

| Property | Value |

+-----------------------+-------------------+

| control\_plane\_version | v1.16.2 |

| hostname | controller-1 |

| id | 2 |

| kubelet\_version | v1.16.1 |

| personality | controller |

| status | upgrading-kubelet |

| target\_version | v1.16.2 |

+-----------------------+-------------------+

$ system host-unlock controller-1

Either controller can be done first.

Upgrade kubelet requires host-lock/unlock.

The kubelets on all controller hosts must be upgraded before upgrading kubelets on worker hosts.

1. Show the kubernetes upgrade status

$ system kube-upgrade-show

+--------------+--------------------------------------+

| Property | Value |

+--------------+--------------------------------------+

| uuid | b5f7dada-2537-4416-9d2c-f9ca9fcd0e22 |

| from\_version | v1.16.1 |

| to\_version | v1.16.2 |

| state | upgrading-kubelets |

| created\_at | 2020-02-20T16:08:48.854869+00:00 |

| updated\_at | 2020-02-20T21:53:16.347406+00:00 |

+--------------+--------------------------------------+

1. Upgrade kubelet on all worker hosts

$ system host-lock worker-1

$ system kube-host-upgrade worker-1 kubelet

+-----------------------+-------------------+

| Property | Value |

+-----------------------+-------------------+

| control\_plane\_version | v1.16.2 |

| hostname | worker-1 |

| id | 3 |

| kubelet\_version | v1.16.1 |

| personality | worker |

| status | upgrading-kubelet |

| target\_version | v1.16.2 |

+-----------------------+-------------------+

$ system host-unlock worker-1

Multiple worker hosts can be upgraded at the same time, as long as there is sufficient capacity remaining on other worker hosts.

1. Complete the kubernetes upgrade

$ system kube-upgrade-complete

+--------------+--------------------------------------+

| Property | Value |

+--------------+--------------------------------------+

| uuid | 4e942297-465e-47d4-9e1b-9fb1630be33c |

| from\_version | v1.16.1 |

| to\_version | v1.16.2 |

| state | upgrade-complete |

| created\_at | 2020-02-19T20:59:51.079966+00:00 |

| updated\_at | 2020-02-24T15:03:34.572199+00:00 |

+--------------+--------------------------------------+