REQ-458 – API Changes – K8splugin
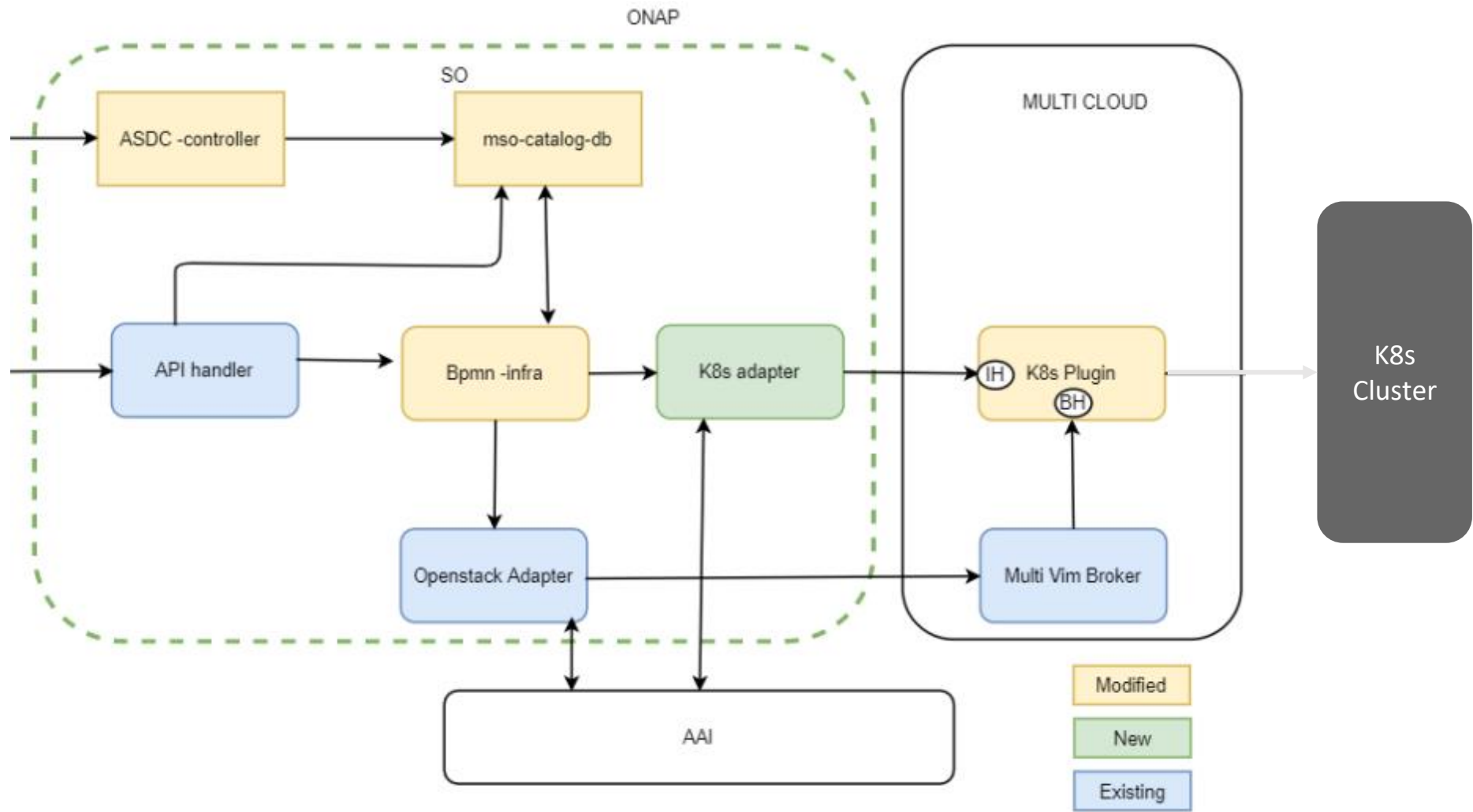
Konrad Bańka (Samsung)

Łukasz Rajewski (Orange)

10.01.2021

Overview

K8splugin usage by SO (Guilin)

K8splugin changes

- K8splugin changes in Honolulu release target "Instance Handler" API set, that is used by SO's K8s Adapter for direct communication with k8splugin. They are not expected to impact existing Multicloud-based communication flow.

- Detailed development is tracked in [MULTICLOUD-1233](#) story

- Created/Updated APIs target:
  - **Adaptation for Day2 Configuration API**
  - **Enhancements for Status API functionalities**
    - Implementation of Query API
    - [Stretch] Implementation of subscription-based Status API
  - **Implementation of Healthcheck API**

API Changes

API Changes – Day2 Config API [Change]

- Day2 Config API is split between Config Template API (anchored to Resource Bundle level) and Config Value API (anchored to Profile level)
  - Only Config Value API will be changed by re-anchoring to Instance level.
  - No other externally-visible changes are expected.
- Removed API Endpoints
  - **POST** /v1/rb/definition/{rbname}/{rbversion}/profile/{prname}/config
  - **GET; PUT; DELETE** /v1/rb/definition/{rbname}/{rbversion}/profile/{prname}/config/{cfgname}
  - **POST** /v1/rb/definition/{rbname}/{rbversion}/profile/{prname}/config/rollback
  - **POST** /v1/rb/definition/{rbname}/{rbversion}/profile/{prname}/config/tagit
- New API Endpoints
  - **POST** /v1/instance/{instance-id}/config
  - **GET; PUT; DELETE** /v1/instance/{instance-id}/config/{cfgname}
  - **POST** /v1/instance/config/{instance-id}/rollback
  - **POST** /v1/instance/config/{instance-id}/tagit

API Changes – Instance Healthcheck API [New]

- Instance Healthcheck API allows utilizing existing Helm's "test" command for k8splugin's managed CNFs (see https://helm.sh/docs/topics/chart_tests/)

- User needs to provide relevant, properly-labeled test jobs in Helm Chart, then this API allows running them against the CNF and review their result

- New APIs Endpoints
  - **GET** /v1/instance/{instance-id}/healthcheck/
    - Returns list of executed tests and their results [Success, Fail, Running]
  - **POST** /v1/instance/{instance-id}/healthcheck
    - Returns ID of started Healthcheck operation to query for its result later
  - **GET** /v1/instance/{instance-id}/healthcheck/{healthcheck-id}
    - Returns status of executed test [Success, Fail, Running] with detailed status of every executed job
  - **DELETE** /v1/instance/{instance-id}/healthcheck/{healthcheck-id}
    - Removes executed test with associated test job in the k8s cluster

API Changes – Query API [New]

- Existing Status API will be extended by possibility to filter-out only relevant resources. This new functionality is served by Query API

- Existing Status API
  - **GET** /v1/instance/{instance-id}/status

- New Query API:
  - **GET** /v1/instance/{instance-id}/query
  - Parameters (**ApiVersion**, **Kind** and either **Name** or **Label** must be provided)
    - **ApiVersion** – string – Resource ApiVersion, example: "v1"; "apps/v1"
    - **Kind** – string – Resource Kind, example: "Pod", "Deployment"
    - **Name** – string – Name of Resource, example: "my-pod-1"
    - **Labels** – string – One or many pairs of label name/value data for filtering Resources, example: "app=test"; "app=test,version=prod"

- Response of Query API uses the same structure as Status API

- Subscription-based Status API allows for subscribing for given RB instance to receive incremental updates about its status change
- Subscription Status API would provide following interfaces (part 1)
  - **POST** /v1/{instance-id}/status/subscriptions
    - Register for instance status
    - Request (encoded in json object):
      - **CallbackURL** – [Mandatory] string – URL to use for sending updates, example "http://dummy.svc:8080/vnf/x/vf-module/y/status/listener"
      - **SimpleAuth** – [Optional] string – Colon-delimited user and password HTTP Basic Auth data to use for CallbackURL, example "user:test"
      - **KeepAliveTime** – [Optional] int – Seconds to hold subscription valid, example "86400"
    - Response (encoded in json object):
      - **SubscriptionID** – string – ID of subscription
      - **KeepAliveTime** – int – Seconds of subscription's validity
  - **GET** /v1/{instance-id}/status/subscriptions
    - Response – list of existing active subscriptions

- Subscription Status API would provide following interfaces (part 2)
  - **GET** /v1/{instance-id}/status/subscriptions/{subscription-id}
    - Responds with details of existing subscription
    - Response
      - **KeepAliveTime** – int – Seconds of current subscription's validity
      - **LastUpdate** – datetime – Datetime of last subscription's update message
  - **POST** /v1/{instance-id}/status/subscriptions/{subscription-id}/refresh
    - **Prolongs** active subscription
    - **Request**
      - **KeepAliveTime** – [Optional] int – Seconds to set leftover subscription validity, example "86400"
    - **Response**
      - **KeepAliveTime** – int – Seconds of updated subscription's validity
  - **DELETE** /v1/{instance-id}/status/subscriptions/{subscription-id}
    - Deletes active subscription

Detailed changes

Adaptation for Day2 Configuration API

- Existing Day2 Configuration API manages Day2 configuration objects on profile, not instance basis. This is due to initial assumption of k8splugin, that single CNF profile should be utilized by a single CNF instance.

- In Guilin this assumption has been loosed, enabling reuse of profile across many CNF instances. This, however, requires now to implement proper adjustments in Day2 Configuration API.

- Further adaptation may be implemented to respect CNF Instance instantiation-time parameters when templating Day2 Configuration objects.

Enhancements for Status API functionalities

- Implemented in Guilin, Status API allows status retrieval of (almost) all objects created and related to CNF Instance.

- To simplify user experience and integration with this API, in Honolulu it is expected to enhance it by accepting query parameters to limit response to selected objects. This feature will either expand Status API endpoint or create a new, dedicated one (Query API).

- **[Stretch]** Implementation of Publish/Subscribe capability of Status API is planned to simplify operation of SO/K8s Adapter for further provisioning of this dynamic inventory information to AAI

Implementation of Healthcheck API

- To support implementation of CNF Healthcheck operation in SO, dedicated Healthcheck (Test) API is planned for implementation.
- This API would utilize capability already available for helm-based deployments - `helm test` - to run predefined set of jobs and retrieve their result
- Jobs utilized by this API would need to be defined in Helm package according to official Helm specification.
- Implementation of this API should also fix current K8splugin behavior of not filtering Test-defined jobs out from instantiation of CNF