2020-05-29 OpenStack Release Selection Meeting Minutes

Meeting Link: https://attcorp.webex.com/join/pg683k

Meeting Number: **739 623 371**

Attendees:

- Name (Organisation)
- Pankaj Goyal
- Cedric Ollivier
- Johanna Heinonen
- Ian Gardner
- Karine Sevilla

Agenda + Minutes

- Agenda Bashing
- Shortlist of candidate releases:
  - Stein and Train
- Comparison of releases as per selection criteria
  - Cyborg API v2.0 support required (China Mobile)
  - Minimise # of releases selected by CNTT and need to frequent upgrades.
    - Train is in the RH LTS program until 2024 – implies that fixes will be available
    - If choose Stein, RA-1 documentation will complete probably by EOY 2020 while Stein will get into extended maintenance by October 2020
  - Ironic support for Redfish (non OSTK) Virtual media boot – Train release.
  - Live migration fixes issues (CPU Pinning, NUMA, SR-IOV) in Train
  - For high performance workloads, Train supports complex trees modeling NUMA layouts, multiple devices, and networks where affinity between and grouping among the members of the tree are required
  - Train: More stable release in terms of the architectural changes
  - Next OPNFV Jerma and CNTT Baraque releases will be aligned on Train.
  - Airship is a set of tools that ingest manifests and helm charts. These files can be modified for any OpenStack release, configurations including API selections
  - Train: possible to create a user with finer-grained access to keystone APIs
  - Train: Application credentials now support access rules, a user-provided list of OpenStack API requests for which an application credential is permitted to be used.
- AOB
- Next Steps:
  - Create documentation and review
  - Submit to Rabi and Walter for TSC meeting (Wednesday) – this can lead to TSC and Governance approvals
  - Create presentation for vF2F

Action Items

	Description	Due Date	Assignee	Status
1	Review Stein and Train functionality and get company perspective	before next meeting	All	
2				
3				

Distros

#	OSTK Release	RHOSP	Ubuntu	Mirantis	Notes
1	Train	16 | LTS | 30 May 2024	Train (on 18.04 LTS) | Feb 2021		RH recommends wait until 16.1 and wait a few months after its GA
2	Stein	15 | | Sep 2020	Stein (on 18.04 LTS) | Mar 2022		
3	Rocky		Rocky (on 18.04 LTS) | Feb 2020		

4	Queens	13 | LTS | 27 Jun 2023	Queens LTS (on 18.04 LTS) | Mar 2023	MCP | EOY 2021	

OSTK Release Notes for Services of interest to CNTT

	Service	Function	Stein: Key Release Notes	Train: Key Release Notes
1	**Barbican** - Key Manager service	To produce a secret storage and generation system capable of providing key management for services wishing to enable encryption features.	• Some enhancements were made to the vault back-end. It is now possible to specify a KV mountpoint and use AppRoles to manage authentication. • We now run a Barbican specific Octavia gate to verify the Octavia load balancing scenario. • The PKCS#11 plugin was modified to allow the hmac_keywrap_mechanism to be configured. With this change, Barbican can be deployed with Ultimaco HSMs. • It is now possible to deploy Barbican with the pkcs#11 backend using either a Thales or an ATOS HSM via TripleO. • Fixes were made to ensure that the barbican-manage commands for key rotation worked for the PKCS#11 plugin.	
2	**Cinder** - Block Storage Service	To implement services and libraries to provide on-demand, self-service access to Block Storage resources via abstraction and automation on top of other block storage devices	• Added multiattach and deferred deletion support for the RBD driver. • Numerous bug fixes have been integrated to address stability and reliability. • User experience improvements around driver initialization, data retained during volume transfers and the information returned by commands. • Continued improvements in the backup service.	• A number of drivers have added support for newer features like multi-attach and consistency groups. • When uploading qcow2 images to Glance the data can now be compressed. • Team focused on getting numerous bug fixes and usability improvements in place. • Cinder now has upgrade checks that can be run to check for possible compatibility issues when upgrading to Train.
3	**Cyborg** - Accelerator Life Cycle Management	To provide a general management framework for accelerators (FPGA, GPU,SoC, NVMe SSD, DPDK/SPDK,eBPF /XDP …)	• Add FPGA programming support • Add GPU drivers • DB re-work to align with NOVA placement api strategy • only v1.0 API	• Got Cyborg-Nova interaction spec merged. This is the blueprint for the end goal, i.e., launching and managing VMs with accelerators. <https://github.com/openstack/nova-specs/blob/master/specs/train/approved/nova-cyborg-interaction.rst> • Updated Cyborg APIs to version 2, which includes support for Nova interaction. Using v2 APIs, end users can create/delete device profiles and create/bind/unbind/delete accelerator requests (ARQs). • Added new Cyborg driver (Ascend) and improved existing drivers (Intel FPGA, GPU). • Created tempest CI framework that can be used with a fake driver today and with real hardware in the future. • Enabled Python 3 testing and fixed issues in support of Train goals. • Supports both v1.0 and v2.0 API
4	**Glance** - Image Service	To provide services and associated libraries to store, browse, share, distribute and manage bootable disk images, other data closely associated with initializing compute resources, and metadata definitions.		• Images API v2.9 has promoted to current with 2.7 & 2.8 marked as SUPPORTED. 2.8 will show in version list only when multi-store is configured. • Glance multi-store feature has been deemed stable • glance-cache-manage is not depending to glance-regstry anymore and is communicating directly with glance-api • Cache prefetching is now done as periodic task by glance-api removing the requirement to add it in cron. • Various bugfixes done in glance, glance_store and python-glanceclient
5	**Heat** - Orchestration Service	To orchestrate composite cloud applications using a declarative template format through an OpenStack-native REST API.	• Heat now supports orchestrating stacks in remote OpenStack clouds, using credentials stored by the user in Barbican. • It is now easier to recover from accidentally trying to replace a resource with a version that conflicts with the existing resource. • New resource types in Heat add support for Neutron Layer 2 Gateways, Blazar, and Tap-as-a-Service. • Support Glance web download image resource type, which allow get the image from URL without pre-load it out side of Glance.	
6	**Horizon** - Dashboard Service	To provide an extensible unified web based user interface for all OpenStack services.	• Cinder Generic Groups admin panels are now supported • Added option to mitigate breach attacks • Added an upgrade_check management command • Custom templates for clouds.yaml and openrc files support	• Volumes multi-attach is supported now • Horizon now supports the optional automatic generation of a Kubernetes configuration file • It is the last release with Python 2.7 and Django 1.11 support
7	**Ironic** - Bare Metal Service	To produce an OpenStack service and associated libraries capable of managing and provisioning physical machines, and to do this in a security-aware and fault-tolerant manner.	• Adds additional interfaces for management of hardware including Redfish BIOS settings, explicit iPXE boot interface option, and additional hardware support. • Increased capabilities and options for operators including deployment templates, improved parallel conductor workers and disk erasure processes, deployed node protection and descriptions, and use of local HTTP(S) servers for serving images. • Improved options for standalone users to request allocations of bare metal nodes and submit configuration data as opposed to pre-formed configuration drives. Additionally allows for ironic to be leveraged using JSON-RPC as opposed to an AMQP message bus.	• Basic support for building software RAID • Virtual media boot for the Redfish hardware type. • Improvements in the sensor data collection. • New tool for building ramdisk images: ironic-python-agent-builder • Numerous fixes in the Ansible deploy interface.

8	**Keystone** - Identity Service	To facilitate API client authentication, service discovery, distributed multi-tenant authorization, and auditing.	• This release introduced Multi-Factor Authentication Receipts, which facilitates a much more natural sequential authentication flow when using MFA. • The limits API now supports domains in addition to projects, so quota for resources can be allocated to top-level domains and distributed among children projects. • JSON Web Tokens are added as a new token format alongside fernet tokens, enabling support for a internet-standard format. JSON Web Tokens are asymmetrically signed and so synchronizing private keys across keystone servers is no longer required with this token format. • Multiple keystone APIs now support system scope as a policy target, which reduces the need for customized policies to prevent global access to users with an admin role on any project. • Multiple keystone APIs now use default reader, member, and admin roles instead of a catch-all role, which reduces the need for customized policies to create read-only access for certain users.	• All keystone APIs now use the default reader, member, and admin roles in their default policies. This means that it is now possible to create a user with finer-grained access to keystone APIs than was previously possible with the default policies. For example, it is possible to create an “auditor” user that can only access keystone’s GET APIs. Please be aware that depending on the default and overridden policies of other OpenStack services, such a user may still be able to access creative or destructive APIs for other services. • All keystone APIs now support system scope as a policy target, where applicable. This means that it is now possible to set `[oslo_policy] /enforce_scope` to `true` in *keystone.conf*, which, with the default policies, will allow keystone to distinguish between project-specific requests and requests that operate on an entire deployment. This makes it safe to grant admin access to a specific keystone project without giving admin access to all of keystone’s APIs, but please be aware that depending on the default and overridden policies of other OpenStack services, a project admin may still have admin-level privileges outside of the project scope for other services. • Keystone domains can now be created with a user-provided ID, which allows for all IDs for users created within such a domain to be predictable. This makes scaling cloud deployments across multiple sites easier as domain and user IDs no longer need to be explicitly synced. • Application credentials now support access rules, a user-provided list of OpenStack API requests for which an application credential is permitted to be used. This level of access control is supplemental to traditional role-based access control managed through policy rules. • Keystone roles, projects, and domains may now be made immutable, so that certain important resources like the default roles or service projects cannot be accidentally modified or deleted. This is managed through resource options on roles, projects, and domains. The `keystone-manage bootstrap` command now allows the deployer to opt into creating the default roles as immutable at deployment time, which will become the default behavior in the future. Roles that existed prior to running `keystone -manage bootstrap` can be made immutable via resource update.
9	**Neutron** - Networking Service	To implement services and associated libraries to provide on-demand, scalable, and technology-agnostic network abstraction.	• Support for strict minimum bandwidth based scheduling. With this feature, Nova instances can be scheduled to compute hosts that will honor the minimum bandwidth requirements of the instance as defined by QoS policies of its ports. • Network Segment Range Management. This features enables cloud administrators to manage network segment ranges dynamically via a new API extension, as opposed to the previous approach of editing configuration files. This feature targets StarlingX and edge use cases, where ease of of management is paramount. • Speed up Neutron port bulk creation. The targets are containers / k8s use cases, where ports are created in groups. • (FWaaS) FWaaS v1 has been removed. FWaaS v2 is available since Newton release and it covers all features in FWaaS v1. A migartion script is provided to convert existing FWaaS v1 objects into FWaaS v2 models.	• OVN can now send ICMP “Fragmentation Needed” packets, allowing VMs on tenant networks using jumbo frames to access the external network without any extra routing configuration. • Event processing performance has been increased by better distributing how work is done in the controller. This helps significantly when doing bulk port bindings. • When different subnet pools participate in the same address scope, the constraints disallowing subnets to be allocated from different pools on the same network have been relaxed. As long as subnet pools participate in the same address scope, subnets can now be created from different subnet pools when multiple subnets are created on a network. When address scopes are not used, subnets with the same `ip_version` on the same network must still be allocated from the same subnet pool. • A new API, extraroute-atomic, has been implemented for Neutron routers. This extension enables users to add or delete individual entries to a router routing table, instead of having to update the entire table as one whole • Support for L3 conntrack helpers has been added. Users can now configure conntrack helper target rules to be set for a router. This is accomplished by associating a conntrack_helper sub-resource to a router.
10	**Nova** - Compute Service	To implement services and associated libraries to provide massively scalable, on demand, self service access to compute resources, including bare metal, virtual machines, and containers.	• It is now possible to run Nova with version 1.0.0 of the recently extracted placement service, hosted from its own repository. Note that install/upgrade of an extracted placement service is not yet fully implemented in all deployment tools. Operators should check with their particular deployment tool for support before proceeding. See the placement install and upgrade documentation for more details. In Stein, operators may choose to continue to run with the integrated placement service from the Nova repository, but should begin planning a migration to the extracted placement service by Train, as the removal of the integrated placement code from Nova is planned for the Train release. • Users can now specify a volume type when creating servers. • The compute API is now tolerant of transient conditions in a deployment like partial infrastructure failures, for example a cell not being reachable. • Users can now create servers with Neutron ports that have quality-of-service minimum bandwidth rules. • Operators can now set overcommit allocation ratios using Nova configuration files or the placement API. • Compute driver capabilities are now automatically exposed as traits in the placement API so they can be used for scheduling via flavor extra specs and /or image properties. • Live migration is now supported for the VMware driver.	• Live migration support for servers with a NUMA topology, pinned CPUs and /or huge pages, when using the libvirt compute driver. • Live migration support for servers with SR-IOV ports attached when using the libvirt compute driver. • Support for cold migrating and resizing servers with bandwidth-aware Quality of Service ports attached. • Improvements to the scheduler for more intelligently filtering results from the Placement service. • Improved multi-cell resilience with the ability to count quota usage using the Placement service and API database. • A new framework supporting hardware-based encryption of guest memory to protect users against attackers or rogue administrators snooping on their workloads when using the libvirt compute driver. Currently only has basic support for AMD SEV (Secure Encrypted Virtualization). • API improvements for both administrators/operators and end users. • Improved operational tooling for things like archiving the database and healing instance resource allocations in Placement. • Improved coordination with the baremetal service during external node power cycles. • Support for VPMEM (Virtual Persistent Memory) when using the libvirt compute driver. This provides data persistence across power cycles at a lower cost and with much larger capacities than DRAM, especially benefitting HPC and memory databases such as redis, rocksdb, oracle, SAP HANA, and Aerospike.
11	**Octavia** - Load Balancer Service	To provide scalable, on demand, self service access to load-balancer services, in technology-agnostic manner.	• Octavia now supports load balancer “flavors”. This allows an operator to create custom load balancer “flavors” that users can select when creating a load balancer. • You can now enable TLS client authentication when using TERMINATED_HTTPS listeners. • Octavia now supports backend re-encryption of connections to member servers. • Metadata tags can now be assigned to the elements of an Octavia load balancer.	• An Access Control List (ACL) can now be applied to the load balancer listener. Each port can have a list of allowed source addresses. • Octavia now supports Amphora log offloading. Operators can define syslog targets for the Amphora administrative logs and for the tenant load balancer connection logs. • Amphorae can now be booted using Cinder volumes. • The Amphora images have been optimized to reduce the image size and memory consumption.
12	**Placement** - Placement Service	To track cloud resource inventories and usages to help other services effectively manage and allocate their resources.	• The placement service was extracted from the Nova project and became a new official OpenStack project called Placement. • Added the ability to target a candidate resource provider, easing specifying a host for workload migration. • Increased API performance by 50% for common scheduling operations. • Simplified the code by removing unneeded complexity, easing future maintenance.	• Train is the first cycle where Placement is available solely from its own project and must be installed separately from Nova. • Extensive benchmarking and profiling have led to massive performance enhancements in the placement service, especially in environments with large numbers of resource providers and high concurrency. • Added support for forbidden aggregates which allows groups of resource providers to only be used for specific purposes, such as reserving a group of compute nodes for licensed workloads. • Added a suite of features which, combined, enable targeting candidate providers that have complex trees modeling NUMA layouts, multiple devices, and networks where affinity between and grouping among the members of the tree are required. These features will help to enable NFV and other high performance workloads in the cloud.

13	**Swift** - Object Storage	Provide software for storing and retrieving lots of data with a simple API. Built for scale and optimized for durability, availability, and concurrency across the entire data set.	• Numerous improvements to the S3 API compatibility layer. • Several fixes and improvements for the data-encryption middleware, including allowing multiple keymaster middlewares. This allows migration from one key provider to another. • Operators have more control over account and container server background daemon I/O usage with the new `databases_per_second` config option. • Erasure-coded data may now be rebuilt to handoff nodes. This improves data durability when disk failures go unremedied for extended periods.	• Swift can now be run under Python 3. • Log formats are now more configurable and include support for anonymization. • Swift-all-in-one Docker images are now built and published to https://hub.docker.com/r/openstackswift/saio
14				
15	**Senlin** - Clustering Service	To implement clustering services and libraries for the management of groups of homogeneous objects exposed by other OpenStack services.	• Improved performance so that Senlin operations execute multiple orders of magnitude faster. • Health policy v1.1 now allows a user to specify multiple types of detection modes. • Senlin APIs now issues synchronous failures in case of cluster/node lock, cooldown in effect or action conflict. • Operators can now remove completed actions using action-purge subcommand in senlin-manage tool. This is useful for long-running clusters that have accumulated a large number of actions in the database.	• Added webhook v2 support:Previously webhook API introduced microversion 1.10 to allow callers to pass arbitrary data in the body along with the webhook call. • Supported admin user can see details of any cluster profile. • Allows the cluster delete actions to detach policies and delete receivers for the cluster being deleted.
16	Ceilomter - Telemetry Service	(or be aligned with RA-2 and utilise Prometheus)		

Cyborg API

	API v1.0	API v 2.0	