LaaS Timeline

- LaaS Community and Governance
 LaaS Evolution: Development Time
 - LaaS Evolution: Development Timeline
 - Laas 1.0 (MVP launched February 2018):
 - LaaS 2.0 (ETA Fall 2018)
 - LaaS 3.0 (Official LaaS offering ETA: "Laas 2.0 + 4-6 weeks"):
 - LaaS 4.0 (Extended LaaS offering):
- Background: Evolving Objectives for LaaS
- References, Previous Discussions & Presentations

LaaS Community and Governance

- LaaS is governed by the OPNFV TSC. Day-2-day business for LaaS is handled by the OPNFV Infra Working Group.
- LaaS is available to all LFN communities, i.e. OPNFV, ONAP, OpenDaylight, FD.io, PNDA, Tungsten Fabric.
- LaaS access policy evolves with the overall evolution of the LaaS program (see details below)

LaaS Evolution: Development Timeline

Laas 1.0 (MVP - launched February 2018):

Product Description:

- Book a single server using the dashboard
- Selected OS installed (Ubuntu, CentOs, Suse)
- · Generate and send out access information automatically

Usage / Uptake:

Typically 50%-80% utilization of servers, with peaks associated to plugfests or other industry events pointing towards the LaaS resources.

Usage Policy

- · Eligible users: Anyone with an LF account
- Number of servers per booking: 1
- Maximum number of concurrent bookings per user: no limit
- Maximum length of a booking: 3 weeks
- · Booking extensions: Yes. Maximum of 2 extensions, 1 week each

LaaS 2.0 (ETA Fall 2018)

Product Description:

- Evolution from MVP (LaaS 1.0): Allow to book multiple bare metal servers, rather than a single one.
- Dynamic POD Allocation
 - $^{\circ}\;$ Allow user to design and book a set of servers with networks between them
 - ° Dashboard will generate PDF for user defined POD
- Smaller features and improvements
 - Multi-user bookings (i.e. adding cohorts to your booked resources, so separate VPN access doesn't have to be requested out of band)
 Allow users to save a "snapshot" or their system(s), and re-deploy that image at a later date
 - ° Support for additional labs in the dashboard.

Usage Policy:

LaaS 2.0 introduces the ability to book a cluster of bare-metal servers. Usage policy differs for single and multi-server bookings.

- Single server bookings (same as LaaS 1.0):
 - Eligible users: Anyone with an LF account
 - Number of servers per booking: 1
 - ° Maximum number of concurrent bookings per user: no limit
 - Maximum length of a booking: 3 weeks
 - Booking extensions: Yes. Maximum of 2 extensions, 1 week each
- Multiple server bookings:
 - Eligible users: Project Technical Leaders (PTL) only. During the booking process, the PTL will need to authorize by providing a pointer to an INFO file in git repo of a LFN project that shows her/him as PTL of that project.
 - Maximum number of servers per booking: 8
 - Maximum number of concurrent bookings per user: no limit
 - Note: We expect that PTLs communicate among each other to ensure that resources are shared fairly. In case this does not work out, a more constrained policy will be put in place.

- Maximum length of a booking: 3 weeks
- Booking extensions: Yes. Maximum of 2 extensions, 1 week each

LaaS 3.0 (Official LaaS offering - ETA: "Laas 2.0 + 4-6 weeks"):

Product Description:

- Automatic deployments (virtual or bare-metal) of OPNFV on top of user defined POD; See also Lab as a Service#LaaSFlowProposal

 Only supported for installers supporting PDF (pod descriptor files)
- Available 4-6 weeks after LaaS 2.0

Usage Policy:

Same as LaaS 2.0.

LaaS 4.0 (Extended LaaS offering):

Product Description:

- Automate installation of other LFN projects
- Available TBD (details are to be defined)

Usage Policy:

Same as LaaS 2.0.

Background: Evolving Objectives for LaaS

- LaaS original objectives as approved by the former OPNFV board:
 - LaaS definition: Lab as a Service#WhatisOPNFVLaaS
 - ° LaaS initial set of use cases: Lab as a Service#OPNFVLaaSinitialuse-cases
 - ^o LaaS deployment approach: Lab as a Service#LaaSFlowProposal
- Evolution Objective #1: Now that OPNFV is part of LFN, start to make LaaS available to the entire LFN
 - Offer LaaS to other communities within the LFN and make it applicable to the other communities in the LFN: ONAP, OpenDaylight, FD.io, PNDA, SNAS, Tungsten Fabric.
 - Several LFN projects (e.g. ONAP, PNDA) are quite resource intense.
 - Example ONAP: OOM install would fit a single server (https://onap.readthedocs.io/en/beijing/guides/onap-developer/settingup /onap_oom.html#installing-onap-k8s) whereas the Heat-based install requires more resources than a single LaaS server can offer (https://onap.readthedocs.io/en/beijing/guides/onap-developer/settingup/onap_heat.html#installing-onap-heat).
 - Example PNDA: http://pnda.io/pnda-guide/provisioning/platform_requirements.html
- Evolution Objective #2: Leverage learnings from initial phase of LaaS ("MVP see below")
- Enable additional use cases (example: LaaS use case: Project Auto), which require a "bare metal" environment of multiple servers.
 Define and manage usage-policies of LaaS:
 - The simple MVP (single server with operating system) is already utilized at 50%. Access restrictions are minimal (LF account is sufficient for LaaS access)
 - Larger community (all LFN) combined with more flexible deployments (one or multiple servers) requires closer and tighter control to ensure that resources are shared fairly.

References, Previous Discussions & Presentations

- 1. Lab as a Service main LaaS wiki page
- 2. OPNFV LaaS Update to the TSC
- 3. LaaS Update during the Plugfest
- 4. LaaS Presentation at ONS