AGENDA

– Introduction to OPNFV Doctor project
– Achievements
– Metrics
OPNFV Doctor project - Introduction

- **Goal:**
  - Develop and build fault management and maintenance framework for high availability of Network Services running on top of virtualized infrastructure.

  ➔ **Proposed with a very clear target / key feature:**
  - Immediate notification of unavailability of virtualized resources from VIM to Consumer

- **Contributing organizations:**
  - NEC (PTL: Ryota Mibu), AT&T, Cisco, Cloudbase Solutions, Corenova, Ericsson, Huawei, Intel, KDDI, KT, Nokia, NTT DOCOMO, Spirent, Sprint, Telecom Italia, ZTE

- **https://wiki.opnfv.org/display/doctor/**
OPNFV Doctor project – Timeline

- **ARNO**
  - Requirement document
  - Ceilometer “Immediate Notification”
  - Nova “Mark Host Down”
  - Functional test cases
  - PoC demo at OPNFV Summit
  - Documentation updates

- **BRAHMAPUTRA**
  - Ceilometer “Immediate Notification”
  - Nova “Mark Host Down”
  - Functional test cases
  - PoC demo at OPNFV Summit
  - Documentation updates

- **COLORADO**
  - Nova: “Get valid server state” and “Add notification for service status change”
  - Integration of Congress as Doctor Inspector
  - Extended functional tests
  - PoC demo at OPNFV Summit and OpenStack Summit
  - Documentation updates
  - OPNFV Plugfest

- **DANUBE**
  - Neutron “Port Status update”
  - Inspector design guidelines
  - Performance profiler
  - Documentation updates
  - OPNFV Plugfest

- **EUPHRATES**
  - Congress: parallel policy action execution for faster fencing, notification and recovery
  - Maintenance specs
  - Code refactoring to Python
  - Collectd as Doctor Monitor

**Timeline events:**
- **OPNFV launch** 30 Sep, 2014
- **Doctor creation** 2 Dec, 2014
- **Arno release** 4 May, 2015
- **OPNFV Summit 2015** 9 Nov, 2015
- **Brahmaputra release** 1 Mar, 2016
- **Danube release** 4 Apr, 2017
- **Colorado release** 16 Sep, 2016
- **Euphrates release** October 2017

**Q4 2015**
- Q1 2015
- Q2
- Q3
- Q4

**Q1 2016**
- Q2
- Q3
- Q4

**Q1 2017**
- Q2
- Q3
- Q4
Workflow
1. Doctor requirements document

http://artifacts.opnfv.org/doctor/docs/development_requirements/index.html

- Use cases and scenarios
  - Active-Standby configuration (1+1 redundancy):
    - Consumer of infrastructure has configured ACT-STBY
    - Fault in virtualized infrastructure (NFVI) → inform the Consumer to switch to STBY instance
  - Prevention actions based on fault prediction: Switch to STBY in case of a predicted fault
  - NFVI maintenance: inform Consumer(s) of affected hardware about planned maintenance

- Requirements
  1. Monitor physical and virtual resources and detect problems
  2. Correlate faults and identify affected virtual resources
  3. Notification of Consumer(s) of affected virtual resources
  4. Execute steps 1-3 in less than e.g. 1 second to avoid service disruption
2. Doctor architecture and integrated (OpenStack) projects

- **Application**
- **Virtualized Infrastructure (Resource Pool)**

- **Manager / Consumer**
  - 0. Set Alarm
  - 5. Notify Error
  - 6. Action

- **Controller**
  - Nova
  - Neutron
  - Cinder
  - Resource Map

- **Monitor**
  - Zabbix
  - collectd

- **Notifier**
  - Aodh
  - Alarm Conf.

- **Inspector**
  - Congress
  - Failure Policy

---

1 OpenStack project
2 Vitrage could be an alternative to Congress
3. Gap analysis and solution brainstorming (examples 1)

**Resource state – missing feature**

To be:
- Nova API shall support to change nova-compute state
- User shall be able to read OpenStack states and trust they are correct

As is (Kilo release):
- When a VM goes down due to a host HW, host OS or hypervisor failure, nothing happens in OpenStack. The VMs of a crashed host/hypervisor are reported to be live and OK through the OpenStack API.
- nova-compute state might change too slowly or the state is not reliable if expecting also VMs to be down.

**Immediate notification – deficiency in operation**

To be:
- VIM to immediately notify unavailability of virtual resource to VIM user.
- User shall only receive fault notifications related to owned resource(s).

As is (Kilo release):
- OpenStack Metering ‘Ceilometer’ can notify unavailability of resource.
- Due to innerworking of Ceilometer (polling of events), notification of faults takes seconds to few minutes.
- Performance issue for Ceilometer in medium to large scale deployments.

**Solution brainstorming:**
- Discussion with experts on best way to address the gap
- Outcome: e.g. BP spec for „mark nova-compute down“
3. Gap analysis and solution brainstorming (examples 2)

Maintenance discussions:
There has been discussions about planned maintenance together with OpenStack operators and with Nova and Craton project. There is yet no complete implementation plan as Nova will not accept the feature inside and operator tool project Craton is lacking contributors.

- OPS session in Austin summit:
  https://etherpad.openstack.org/p/AUS-ops-Nova-maint

- OPS session in Barcelona summit:
  https://etherpad.openstack.org/p/BCN-ops-informal-meetup

- Ops sessions Milan mid-cycle summit:
  https://etherpad.openstack.org/p/MIL-ops-telco-nfv
  https://etherpad.openstack.org/p/MIL-ops-inventory-and-fleet-management

- OpenStack Nova Blueprint:
  https://blueprints.launchpad.net/nova/+spec/maintenance-reason-to-server
4. Test cases and user manual

- End to end test cases
  - **Upstream**: unit tests and scope-restricted functional tests upstream
  - **Downstream**: E2E functional tests will validate full systems integration

- Manuals
  - **Requirement** and **design** documents
  - **User guide**, **config guide**, **API guide**
  - How to use **implemented blueprints**
  - How to run the tests and interpret results
  - Doctor project “Solution brief”

---

```
gerit.openstack.org/go/vst/doc
debug
eax
/tv
```
5. PoCs, demos and hackfests

- Keynote demo at OpenStack Barcelona 2016
- PoC at OPNFV Summit 2016
- Meetup at OPNFV Summit 2015
- PoC at OPNFV Summit 2017
- Demo at OPNFV Summit 2015
## 6. Upstream achievements

<table>
<thead>
<tr>
<th>Project</th>
<th>Blueprint</th>
<th>Spec Drafter</th>
<th>Lead Developer</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aodh</td>
<td>Event Alarm Evaluator</td>
<td>Ryota Mibu (NEC)</td>
<td>Ryota Mibu (NEC)</td>
<td>Completed (Liberty)</td>
</tr>
<tr>
<td>Nova</td>
<td>New nova API call to mark nova-compute down</td>
<td>Tomi Juvonen (Nokia)</td>
<td>Roman Dobosz (Intel)</td>
<td>Completed (Liberty)</td>
</tr>
<tr>
<td></td>
<td>Support forcing service down</td>
<td>Tomi Juvonen (Nokia)</td>
<td>Carlos Goncalves (NEC)</td>
<td>Completed (Liberty)</td>
</tr>
<tr>
<td></td>
<td>Get valid server state</td>
<td>Tomi Juvonen (Nokia)</td>
<td>Tomi Juvonen (Nokia)</td>
<td>Completed (Mitaka)</td>
</tr>
<tr>
<td></td>
<td>Add notification for service status change</td>
<td>Balazs Gibizer (Ericsson)</td>
<td>Balazs Gibizer (Ericsson)</td>
<td>Completed (Mitaka)</td>
</tr>
<tr>
<td>Congress</td>
<td>Push Type Datasource Driver</td>
<td>Masahito Muroi (NTT)</td>
<td>Masahito Muroi (NTT)</td>
<td>Completed (Mitaka)</td>
</tr>
<tr>
<td></td>
<td>Adds Doctor Driver</td>
<td>Masahito Muroi (NTT)</td>
<td>Masahito Muroi (NTT)</td>
<td>Completed (Mitaka)</td>
</tr>
<tr>
<td>Neutron</td>
<td>Port data plane status</td>
<td>Carlos Goncalves (NEC)</td>
<td>Carlos Goncalves (NEC)</td>
<td>Completed (Pike)</td>
</tr>
</tbody>
</table>
Metrics

268  # changesets
17   # changeset submitters
Future plans

● Fault management:
  ○ Extend framework with automated failure handling / healing
  ○ Work with proposed OpenStack SIG
    https://etherpad.openstack.org/p/self-healing-rocky-forum

● Maintenance
  ○ Keep working on maintenance topics
    in cooperation with OpenStack operators, Nova, Craton, ...
Summary

• Describe the problem being solved by project @ Project creation
  – Lack of fault detection, notification and recovery mechanism in OpenStack
  – OpenStack's inability in receiving and executing maintenance instructions
  – Requirements shall be produced to solve the problems above

• Project requirements
  1. Monitor physical and virtual resources and detect problems/planned maintenance
  2. Correlate faults and identify affected virtual resources
  3. Notification of Consumer(s) of affected virtual resources
  4. Execute steps 1-3 in less than e.g. 1 second to avoid service disruption
THANKS