ONAP: Stability, Resiliency and Stress tests

Morgan Richomme, Orange

@morganrichomme
ONAP: Stability, Resiliency and Stress tests

• Agenda
  - Reminders
  - Current status
  - Goals
Reminders
ONAP: Stability, Resiliency and Stress tests

ONAP stability/resiliency/stress tests
What are we talking about?

Natacha Chéreau (Orange), Morgan Richomme (Orange)

February, 2020
ONAP: Stability, Resiliency and Stress tests

Critical questions

- Can ONAP survive the restart of a POD? **most of the time yes, it depends on the pod.**
- Can ONAP survive a helm chart redeployment? **most of the time yes but not 100% guarantee**
- Can ONAP survive the restart of a k8s controller? **Not sure, as ONAP is not cloud native yet, the full promise of k8s is not met yet.**
- Can ONAP survive the restart of a k8s worker? **not sure especially if some DBs are on this host. One more time the promise of k8s is not met.**
- Do we have figures on an ONAP solution for a given lab: number of simultaneous onboarding/instantiation/loop? **not really**
- Can we perform a Backup&Restore (disaster scenario)? **maybe with pain**
- Can we perform a smooth upgrade? **clearly no**
- Can the community version be used in production? **depend how brave you are but reasonably not yet**
Current status
Stability and resiliency tests now integrated in weekly CI

- **Stability**
  - Onboarding testing: 10 // service onboarding during 24h
  - Instantiation testing: 10 // service instantiation during 24h

- **Resiliency**
  - Node drain test based on Litmus
ONAP: Stability, Resiliency and Stress tests

ONAP Xtesting results

70% LAST CHECK | ISTANBUL ONAP VERSION
70% TREND | ORANGE LAB OWNER

- infrastructure-healthcheck: 2021-11-14 09:19
  - 20% LAST CHECK
  - Full results

- healthcheck: 2021-11-14 09:19
  - 90% LAST CHECK
  - Full results

- smoke usecases: 2021-11-14 09:19
  - 89% LAST CHECK
  - Full results

- security: 2021-11-14 09:19
  - 100% LAST CHECK
  - Full results

- info only: 2021-11-14 09:19
  - Results
  - Note: results may be uploaded a day after the page is created, be patient...

- stability
  - Stability results
    - 10://basic_van_24h
    - 5://basic_vanboard_24h

- resiliency
  - Resiliency results
  - Note: results may be uploaded after several days, be patient...

[Diagram showing ONAP Xtesting results]
ONAP: Stability, Resiliency and Stress tests

- Honolulu: optimization SDC/Cassandra backend
- Istanbul: regressions detected (JIRA SDC-3773) during first campaign, quickly fixed by SDC team
- We (integration / no feedback from TSC/PTL) consider that the results is acceptable 1000 services onboarded with a success rate of 90% in less than 24h
- The linear increase is understandable as we cannot clean the resources by design (a model once created cannot be deleted to ensure consistency), so the more you onboard the more objects you have to deal with. The test includes a step when we retrieve all the models to detect if the model already exists or not.

Basic onboard

- Number of tests: 1085
- Global success rate: 92%
- Number of simultaneous tests: 5
- Test duration: 1440 m
- Number of executed series: 217
- Min duration: 111
- Max duration: 799
- Mean duration: 366
Honolulu: Fixes done by SO team

Istanbul: Number of simultaneous instances increased => issue when MariaDB replica is set to 3 => recommendation to move to 1. Very good results when replicaset set to 1 (success rate above 95%). However errors occur after 35h (timeout) JIRA created SO-3791 in Istanbul, wait for SO team feedback.

- We (integration) consider that it is not blocking but further investigation should be done
  - Understand the nature of the errors after 35h (dimensioning issue?)
  - Optimize SO/Camunda/Mariadb back end
  - Reduce the linear test duration increase (could ne linked to item 1)

- Unlike for onboarding, we use here only 1 model and we clean all the resources in the SO and in the target Openstack. We should not observe such drift (JIRA created in Frankfurt for this issue)
ONAP: Stability, Resiliency and Stress tests

- Litmus = k8s chaos engineering framework
- Node drain consisting in evicting a full node is working well BUT the evicted components are the ones scheduled randomly on the first node (~ 15 pods among the 200)
- Node memory and cpu hog should be tuned to be more useful
- Lots of Litmus scenarios not used even if they would be applicable to ONAP
ONAP: Stability, Resiliency and Stress tests

- Stability and resiliency tests can be triggered in weekly chain
- BUT
  - weekly chains are no more triggered weekly (not enough resources to install a dedicated ONAP + humane resources to exploit the results)
  - Analysis and troubleshooting is complex and require cross projects work (Integration/SDC, Integration/SO)
  - On instantiation SO is the main contact but the issue is not necessarily in SO...
Goals
ONAP: Stability, Resiliency and Stress tests

- **Stability**
  - Improve the understanding on the linear growth of the instantiation duration
  - Increase the test duration from 24h to 1 week
  - Better tune the Databases
  - Use new test scenario (stimulate more CDS, SDNC,..)
• Resiliency
  - Improve the node drain test to detect components that are not “cloud native” enough to support such test – currently the choices of the components is random, all the components scheduled on the first compute node
  - Improve the use of Litmus: the framework is there and only a small part of the tests are exploited
    • Tune Node CPU/Memory heating
    • Use Cassandra and MariaDB chaos scenarios
ONAP: Stability, Resiliency and Stress tests

• Stress
  – Automate a stress campaign on AAI
    • to find dimensioning figures (number of objects, response time, ..)
    • levering ATT scripts based on gatling (https://gatling.io/)
Conclusions
Can ONAP survive the restart of a POD? most of the time yes
Can ONAP survive a helm chart redeployment? most of the time yes
Can ONAP survive the restart of a k8s controller? yes
Can ONAP survive the restart of a k8s worker? most of the time yes

Do we have figures on an ONAP solution for a given lab: number of simultaneous onboarding/instantiation/loop? **We found some figures for onboarding and instantiation**

Can we perform a Backup&Restore (disaster scenario)? still with pain
Can we perform a smooth upgrade? **Still not**
Can the community version be used in production? We have a much better view than in 2020 conservative organizations would probably say not yet but not only for stability/resiliency reasons (security/versioning,B&R). Additional work is needed (reduction of the component scope (exclude unmaintained components), hardening of the kubernetes/OS, completion of k8s with additional services (storage class, vault, supervision stack,..)
Questions