DefCore Board Report

Meeting May 17, 2015
Rob Hirschfeld & Egle Sigler Co-Chairs

20 minutes

- Process 2015A Discussion [5 mins. **WARNING MINIMAL TIME]**
- 2015.04/03 Update [5 mins **WARNING MINIMAL TIME**]
- 2015.05 Guideline Approval [5 mins]
- Next Cycle Objectives [5 mins]

We are asking the board to approve Process 2015A & 2015.05 Guidelines

DefCore 2015A Process Review & Approval

WARNING MINIMAL DISCUSSION TIME

Board members are expected to have read > http://bit.ly/defcore2015a (Duplicated in appendix of the report for completeness.)

Resolution: Board approves the DefCore 2015A process captured in http://git.openstack.org/cgit/openstack/defcore/tree/process/2015A.rst identified by hash I0bd4ee4a3de4ff89b259dca607ba05cb8d0b7c6f.

DefCore 2015.04/03 Update

- Celebrate DefCore Passing Vendors!
    - Expecting 8 to 12 Vendors
    - Foundation to provide list at meeting (do not want to pre-announce)
- Overall, process seems to be working
- Refstack Up and Working: http://refstack.net
- Starting to see requests for Flagged Tests
    - Flags make sense
    - Flags reflect real use of DefCore 2015.04/03

DefCore 2015.05 Guideline Approval

WARNING MINIMAL DISCUSSION TIME

See Appendix

Noteworthy Changes:

- Keystone Added
    - Required in Compute
    - Advisory in Object
- Tentative Schema change
    - Improves identification of tests
    - Makes it easier to track flagged tests
- Flagged Tests!

Resolution: Board approves the DefCore 2015.05 Guideline process captured in http://git.openstack.org/cgit/openstack/defcore/tree/2015.05.json identified by hash la1523cb0ee4ecd6ae951303663e8afbbfbc9a4e2.

Next Cycle Objectives

- FEWER MEETINGS (regular schedule, perhaps bi-weekly or less)
- Merge Meetings (Capabilities will be "normal agenda")
- Need to clarify Board membership on roster
    - visit: http://wiki.openstack.org/wiki/Governance/DefCoreCommittee
- Hot Topics for "Flag" Cycle
    - Tests outside of Tempest
    - Overlapping APIs (optional components in platform)
    - Implementation vs API (Go in Swift?)

Appendix

2015.05 Guideline https://review.openstack.org/#/c/180376/

OpenStack DefCore 2015.05

Status:	approved
Replaces	2015.04

This document outlines the mandatory capabilities and designated sections required to exist in a software installation in order to be eligible to use marks controlled by the OpenStack Foundation.
This document was generated from the master JSON version.

Releases Covered

Applies to Icehouse, Juno, Kilo

Platform Components

Required:	Compute, Object
Advisory:	None
Deprecated:	None
Removed:	None

Compute Component Capabilities

Required Capabilities

- Identity-auth (keystone)
- Compute-auth (nova)
- Compute-flavors (nova)
- Compute-images (nova)
- Compute-instance-actions (nova)

- Compute-keypairs (nova)
- Compute-quotas (nova)
- Compute-servers (nova)
- Compute-volume (nova)
- Images-v2 (nova)

Advisory Capabilities

- Auth-token (keystone)
- Compute-servers-metadata (nova)

Deprecated Capabilities

None

Removed Capabilities

None

Object Component Capabilities

Required Capabilities

- Objectstore-object (swift)

Advisory Capabilities

- Identity-auth (keystone)
- Auth-token (keystone)

Deprecated Capabilities

None

Removed Capabilities

None

Designated Sections

The following designated sections apply to the same releases as this specification.

Required Designated Sections

- Cinder : Designated sections are the API implementation code
- Glance : Designated sections are the API implementation code and domain model.
- Keystone : Designation is outlined per API grouping. Identity (user and group) management APIs will not be designated. API access (with exception of auth) may be prohibited by policy (resulting in HTTP 403). Designated APIs include both v2.0 and v3 versions where applicable.
- Nova : By default, designated except scheduler, filter, drivers, API extensions and networking.
- Swift : Designated sections are proxy server, object server, container server, account server and select middleware

Advisory Designated Sections

- Keystone : Update pending from PTL

Deprecated Designated Sections

None

Removed Designated Sections

None

OpenStack DefCore Process 2015A (from bit.ly/defcore2015a)

Status: Draft Replaces: none

This document describes the DefCore process required by the OpenStack bylaws and approved by the OpenStack Technical Committee and Board.

Expected Time line:

Time Frame	**Milestone**	**Activities**	**Lead By**
-3 months	S-3	"Preliminary” draft (from current)	DefCore
-2 months	S-2	ID new Capabilities	Community
-1 month	S-1	Score Capabilities	DefCore
Summit	S	"Solid" draft	Community
		Advisory/Deprecated items selected	DefCore
+1 month	S+1	Self-testing	Vendors
+2 months	S+2	Test Flagging	DefCore
+3 months	S+3	Approve Guidance	Board

Note: DefCore may accelerate the process to correct errors and omissions.

Process Definition

The DefCore Guideline process has two primary phases: Draft and Review. During the Draft phase (A), the DefCore Committee is working with community leaders to update and score the components of the guideline. During the Review phase (B), general community and vendors have an opportunity to provide input and check the guidelines (C) against actual implementations. Review phase ends with Board approval of the draft guideline (D).

This section provides specific rules and structure for each phase.

NOTE: To ensure continuity of discussion, process components defined below must _not_ reuse numbers in future revisions. The numbering pattern follows draft, section and sub-item numbering, e.g.: 2015A.B2.2. This requirement may create numbering gaps in future iterations that will help indicate changes.

Guidelines Draft Phase (A)

Starting: S-3

A1. New Guidelines Start From Previous Guidelines

1. *New Guidelines start from the previous Board approved document.*
2. *New Guidelines are given the preliminary name of the target year and .next. (ref section D4)*

A2. Community Groups Tests into Capabilities

1. *DefCore Committee coordinates community activities with the Technical Leadership to revise the capabilities based on current technical needs and functionality.*
2. *Capabilities must correspond to projects which are part of the "TC-approved release" as designated by the TC (see bylaws of the Foundation, section 4.1(b)(iii)).*
3. *Groupings may change between iterations.*
4. *Tests must have unique identifiers that are durable across releases and changes in grouping.*
5. *Tests must be under OpenStack Technical Committee governance.*

6. *The DefCore committee will provide the test groupings in JSON format for scoring.*
7. *The DefCore committee will provide a human-readable summary of the Guideline generated from the JSON version.*

A3. DefCore Collects Recommendations for Designated Sections

1. *Designated Sections will not be removed without being deprecated in the previous Guideline.*
2. *Designated Sections will not be added without being advisory in the previous Guideline.*
3. *Designated Sections will not be added or be made advisory unless the corresponding code base is designated as part of the "TC-approved release" by the Technical Committee (see bylaws of the Foundation, section 4.1(b)(iii)).*
4. *Technical leadership may, but is not required to, assist DefCore with defining advisory sections for projects that have advisory or required capabilities.*
5. *Designated Sections may be sufficiently defined for Guidelines using general descriptions.*
6. *DefCore will present A3.4 descriptions to the Board for approval.*
7. *Technical leadership may, but is not required to, provide more specific details describing the Designated Sections for a project.*
8. *Designated Sections will be included in the JSON Guideline.*

A4. DefCore Committee identifies required capabilities

1. *DefCore uses Board approved DefCore scoring criteria to evaluate capabilities.*
2. *DefCore needs Board approval to change scoring criteria.*
3. *Scoring criteria factor or weights cannot change after Draft is published.*
4. *DefCore identifies cut-off score for determining that a capability is required.*

5. *Capabilities will not be removed without being deprecated in the previous Guideline.*
6. *Capabilities will not be added without being advisory in the previous Guideline.*

A5. Foundation Staff recommends OpenStack Components and OpenStack Platform Scope

1. *Foundation Staff recommends capabilities to include in each OpenStack Component.*
2. *Foundation Staff recommends which Components are required for the OpenStack Platform.*

A6. Additional Capabilities and Tests

1. *DefCore will work with the community to define new capabilities.*
2. *Test grouping for new capabilities will be included in the DefCore documents.*
3. *DefCore will publish a list of missing capabilities and capabilities with inadequate test coverage.*

A7. DefCore Committee creates recommendation for Draft.

1. *DefCore Committee coordinates activities to create draft.*
2. *DefCore Committee may choose to ignore recommendations with documented justification.*

Guidelines Review Phase (B)

Starting: Summit

B1. All Reference Artifacts are reviewed via Gerrit

1. *Draft Guideline*

2. *Designated sections*
3. *Test-Capability groupings*
4. *Flagged Test List*
5. *Capability Scoring criteria and weights*
6. *Not in Gerrit: Working materials (spreadsheets, etc)*

B2. Presentation of Draft Guidelines for Review

1. *DefCore will present Draft Guidelines to the Board for review.*
2. *DefCore will distribute Draft Guidelines to the community for review.*
3. *Foundation Staff will provide Draft Guidelines to vendors for review.*
4. *A link to the Gerrit document must be provided with the review materials.*

B3. Changes to Guideline made by Gerrit Review Process

1. *Community discussion including vendors must go through Gerrit.*
2. *All changes to draft must go through Gerrit process.*
3. *DefCore will proxy for users who do not participate in the Gerrit process with attribution.*

B4. For Gerrit reviews, DefCore CoChairs act as joint PTLs

1. *Board committee members of DefCore serve as "core" reviewers (+2).*
2. *Requests for changes must be submitted as patches by the requesting party.*
3. *DefCore Committee members may proxy change requests as long as the requesting party is explicitly acknowledged.*

Community Review & Vendor Self-Test (C)

Starting: S and continues past S+3

C1. Vendor Self-Tests

1. *Vendors are responsible for executing tests identified by the DefCore committee.*
2. *The Foundation may, but is not required to, provide tooling for running tests.*
3. *The Foundation may, but is not required to, define a required reporting format.*
4. *Self-test results may be published by Vendors in advance of Foundation review, but must be clearly labeled as "Unofficial Results - Not Yet Accepted By The OpenStack Foundation".*
5. *Vendors who publish self-tests MUST provide them in the same format that would be submitted to the OpenStack Foundation but MAY provide additional formats if they choose to do so.*
6. *Self-test results cannot be used as proof of compliance.*

C2. Vendor submits results to Foundation for review

1. *The Foundation determines the acceptable format for submissions.*
2. *The Foundation has final authority to determine if Vendor meets criteria.*
3. *The Foundation will provide a review of the results within 30 days.*

C3. Vendor Grievance Process

1. *Vendors may raise concerns with specific tests to the DefCore committee.*
2. *The DefCore committee may choose to remove tests from a Guideline (known as flagging).*
3. *The DefCore committee will acknowledge vendor requests to flag tests within 30 days.*
4. *Vendors may not request flagging all tests in a capability.*
5. *A reason for the flag, which may include but is not limited to*
    1. *a capability not being widely supported or an existing bug in the test.*

    i. *A remediation step to remove the flag. Options include removing the test from the next release, fixing an existing feature in upstream code, or fixing the bug in the test suite.*
    ii. *A date of test flagging to give guidance to the committee for future test removal.*

6. *Vendors must make a good-faith effort to document the reason for flagging, and where possible attempt to fix upstream bugs in test or production code before submitting tests for flagging.*

C4. Results of Vendor Self-Tests will be open

1. *The Foundation will make the final results of approved vendors available to the community.*
2. *The Foundation will not publish incomplete or unapproved results.*
3. *Only "pass" results will be reported. Skipped and failed results will be omitted from the reports.*
4. *Reports will include individual test results, not just capability scoring.*

C5. API Usage Data Report

1. *The Foundation will provide DefCore committee with an open report about API usage based on self-tests.*
2. *To the extent the data is available, capabilities beyond the DefCore list will be included in the report.*

Guideline Approval (D)

Starting: S+3

D1. Board will review and approve DefCore Guideline from draft

1. *Guidelines are set at the Platform, Component and Capability level only.*
2. *The DefCore Committee will submit the human-readable summary of capabilities (see section A2[6]) to the Board for approval.*
3. *By voting to approve the summary, the Board delegates responsibility for maintaining test groupings to the DefCore committee subject to the limitations described in section D2.*
4. *Guidelines only apply to the identified releases (a.k.a. release tags).*

D2. DefCore Committee has authority on test categorization

1. *DefCore Committee can add flagged tests before and after Guideline approval.*
2. *DefCore Committee cannot add additional Tests to Capability mappings after approval.*
3. *DefCore Committee maintains the test to capability mappings in the JSON representation.*

D3. Designated sections only enforced for projects with required capabilities

1. *Designated sections may be defined for any project.*
2. *Designated sections apply to the releases (a.k.a. release tags) identified in the Guideline.*
3. *Designated sections will be included in the JSON Capabilities file to ensure a single source of identification.*

D4. Guidelines are named based on the date of Board approval

1. *Naming pattern will be: 4-digit year, dot (period), and 2-digit month.*