Building robust applications on top of OpenDaylight

Stephen Kitt, Red Hat

LINUX FOUNDATION
COLLABORATIVE PROJECTS

Robust?

Well-tested

Re-tested

Automatically tested

Automated tests:
tests carried out without human intervention

More automated tests → more robust applications?

More automated tests → more time for manual tests!

More manual tests → more automated tests

More automated tests → fewer regressions

Automated tests in OpenDaylight

- Unit tests
- Integration tests
- CSIT

Unit tests

Small?

Single behaviour

Close to the code...

… but not tied to the implementation

Side benefits of unit tests

- “Implementation” of a specification
- API smoke tests

Unit tests in OpenDaylight

Each project's responsibility

~30,000 unit tests

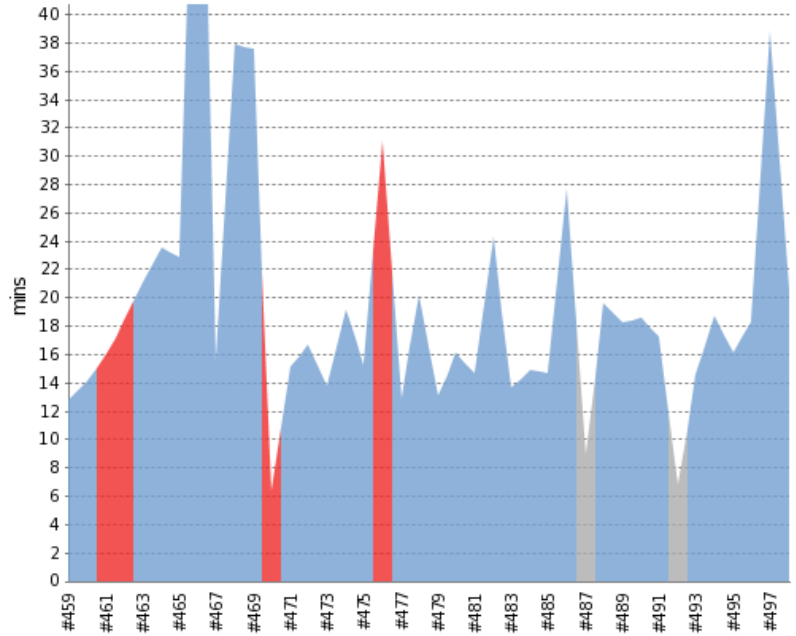
Run for every commit
(submitted for review)

(but only the affected project's tests)

Build History		trend
#498	14-Jun-2016 19:53	40321,1
#497	14-Jun-2016 14:33	39917,5
#496	14-Jun-2016 13:29	40261,1
#495	14-Jun-2016 01:33	40273,1
#494	14-Jun-2016 00:55	39917,5
#493	14-Jun-2016 00:08	39988,3
#492	14-Jun-2016 00:00	39988,3
#491	13-Jun-2016 23:06	40261,1
#490	13-Jun-2016 22:14	39988,3
#489	13-Jun-2016 21:21	39917,5
#488	13-Jun-2016 21:17	39988,3
#487	13-Jun-2016 21:08	39988,2
#486	13-Jun-2016 17:40	39917,4
#485	13-Jun-2016 10:10	40223,1
#484	11-Jun-2016 00:36	39988,1
#483	10-Jun-2016 20:20	39801,4
#482	10-Jun-2016 19:07	39801,4
#481	10-Jun-2016 18:51	39801,4
#480	10-Jun-2016 18:48	39917,4
#479	09-Jun-2016 21:12	39917,3

Build Time Trend

Build ↑	Duration	Slave
#498	20 min	rk-c7-matrix-88
#497	38 min	rk-c7-matrix-88
#496	18 min	rk-c7-matrix-88
#495	16 min	rk-c7-matrix-88
#494	18 min	rk-c7-matrix-88
#493	14 min	rk-c7-matrix-88
#492	6 min 46 sec	rk-c7-matrix-88
#491	17 min	rk-c7-matrix-88
#490	18 min	rk-c7-matrix-88
#489	18 min	rk-c7-matrix-88
#488	19 min	rk-c7-matrix-88
#487	8 min 56 sec	rk-c7-matrix-88
#486	27 min	rk-c7-matrix-88
#485	14 min	rk-c7-matrix-88
#484	14 min	rk-c7-matrix-201
#483	13 min	rk-c7-matrix-201
#482	24 min	rk-c7-matrix-201
#481	14 min	rk-c7-matrix-201
#480	16 min	rk-c7-matrix-201
#479	13 min	rk-c7-matrix-201
#478	20 min	rk-c7-matrix-201
#477	12 min	rk-c7-matrix-201



OPENDAYLIGHT

Required to pass before merge

Code-Review +2 Sam Hague
Verified +1 jenkins-releng

Author	Patrick Roanhouse <Patrick.M.Roanhouse@intel.com>	14/06/2016 20:43
Committer	Anil Vishnoi <vishnoianil@gmail.com>	14/06/2016 22:19
Commit	9a022bca7a1fcbf399b0ed8eee3880ef37d5ce12	(gitweb)
Parent(s)	a03490a4949542cee325ad8186a32b8d8c38733a	(gitweb)
Change-Id	Ia3afae16031f1d43e5c4ccfc54913d636cf4facb	

Files Open All Diff against: Base

File Path	Comments	Size
Commit Message		
hwvtepsouthbound/hwvtepsouthbound-impl/src/main/java/org/opendaylight/ovsdb/hwvtepsouthbound/HwvtepSouthboundUtil.java	2	
		+0, -2

History Expand All

Anil Vishnoi	Patch Set 1: Cherry Picked from branch master.	14/06 22:19
jenkins-releng	Patch Set 1: Build Started https://jenkins.opendaylight.org/releng/job/ovsdb-distribution-check-beryllium/93/ (1/2)	14/06 22:19
jenkins-releng	Patch Set 1: Build Started https://jenkins.opendaylight.org/releng/job/ovsdb-verify-beryllium/2277/ (2/2)	14/06 22:19
jenkins-releng	Patch Set 1: Verified+1 Build Successful https://jenkins.opendaylight.org/releng/job/ovsdb-verify-beryllium/2277/ : SUCCESS https://jenki...	14/06 22:49
Sam Hague	Patch Set 1: Code-Review+2	03:17
Gerrit Code Review	Change has been successfully merged by Sam Hague	03:17
jenkins-releng	Patch Set 1: Build Started https://jenkins.opendaylight.org/releng/job/ovsdb-merge-beryllium/983/	03:17

jenkins-releng 04:01↩

Patch Set 1:

Build Successful

https://jenkins.opendaylight.org/releng/job/ovsdb-merge-beryllium/983/ : SUCCESS

OPENDAYLIGHT

Tracked in Sonar

ovsdb

11 June 2016 14:42 Version 1.3.0-SNAPSHOT

Technical Debt | Coverage | Duplications | Structure | Dashboards | Code | Issues

Coverage

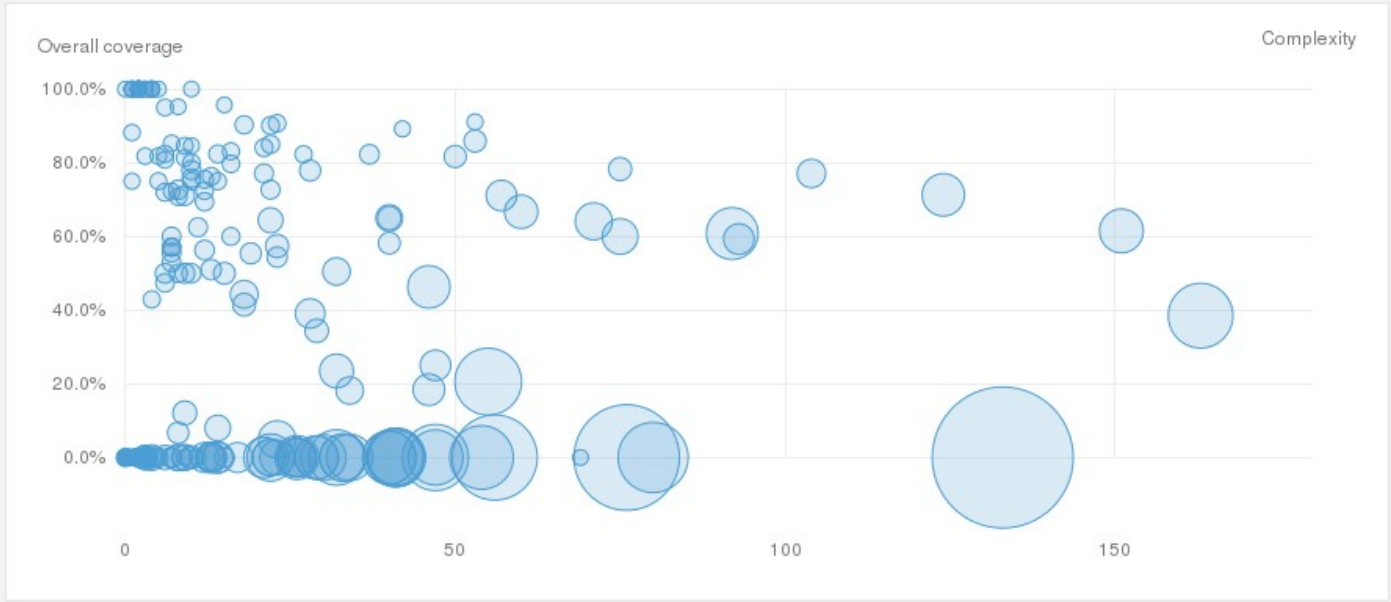
Leak Period: since 1.2.1-SNAPSHOT started 5 months ago

Overall coverage	38.4%	+6.5%
Overall line coverage	41.4%	+6.4%
Overall uncovered lines	6,046	-8.2k
Lines to cover	10,321	-12k
Overall condition coverage	31.0%	+7.2%
Overall uncovered conditions	2,883	-3.3k
Conditions to cover	4,178	-4k
Overall coverage on new code	39.7%	

Coverage	9.8%	-0.8%
Line coverage	11.0%	-0.4%

Files

Size: Overall uncovered lines



OPENDAYLIGHT

Integration tests

Integrating multiple behaviours

Tests closer to end users' expectations
(use cases)

Side benefits of integration tests

- “Implementation” of a use case
- More API smoke tests

Integration tests in OpenDaylight

Each project's responsibility

~300 integration tests

Some requiring non-automated setup

Most run for every commit

Most required to pass before merge

Not distinguished from unit tests in Gerrit

Tracked separately in Sonar

IT coverage	59.0%	+22.6%
IT line coverage	63.5%	+22.5%
IT uncovered lines	2,289	-7.1k
Lines to cover	10,321	-12k
IT condition coverage	47.7%	+23.1%
IT uncovered conditions	1,319	-3.4k
Conditions to cover	4,178	-4k
Coverage by IT on new code		59.9%

CSIT

Continuous System Integration Tests

Integration tests with external requirements

Full-blown system tests: clustering, scalability, longevity...

CSIT in OpenDaylight

Release commitment

- At least one system test for each OSGi feature
- Large-scale system tests for
  - Functionality
  - Clustering
  - Scalability
  - Performance
  - Longevity/stability

Managed by each project **and** the Integration/Test project

~4,000 system tests

Tests run daily
(no gating)

ovsdb-csit-1node-southbound-all-beryllium	8 hr 55 min - #1015	N/A	16 min	N/A	-	102 / 102 passed
ovsdb-csit-1node-southbound-all-boron	1 day 0 hr - #216	10 days - #207	12 min	N/A	-	99 / 102 passed
ovsdb-csit-1node-southbound-all-stable-lithium	16 hr - #755	9 days 18 hr - #741	17 min	N/A	-	99 / 102 passed
ovsdb-csit-1node-southbound-only-beryllium	7 hr 8 min - #1162	2 days 21 hr - #1155	16 min	N/A	-	102 / 102 passed
ovsdb-csit-1node-southbound-only-boron	23 min - #433	8 days 14 hr - #409	19 min	N/A	-	102 / 102 passed
ovsdb-csit-1node-southbound-only-stable-lithium	2 days 12 hr - #701	26 days - #671	16 min	N/A	-	98 / 102 passed
ovsdb-csit-3node-clustering-only-beryllium	7 hr 2 min - #410	2 days 21 hr - #405	22 min	N/A	-	156 / 156 passed
ovsdb-csit-3node-clustering-only-boron	4 hr 4 min - #398	8 days 14 hr - #376	40 min	N/A	-	156 / 156 passed
ovsdb-csit-3node-clustering-only-stable-lithium	2 days 12 hr - #110	N/A	42 min	N/A	-	151 / 156 passed
ovsdb-csit-verify-1node-southbound	1 day 12 hr - #550	21 days - #543	8 min 55 sec	N/A	-	99 / 102 passed
ovsdb-csit-verify-3node-clustering	1 day 12 hr - #379	21 days - #372	17 min	N/A	-	156 / 156 passed

Most tests use Robot Framework

http://robotframework.org/

[integration/test.git] / csit / suites / ovsdb / Southbound_Cluster / 010__Ovsdb_Southbound_Cluster.robot

```
*** Settings ***
Documentation     Test suite for Ovsdb Southbound Cluster - Owner failover and recover
Suite Setup       Create Controller Sessions
Suite Teardown    Delete All Sessions
Library           RequestsLibrary
Resource          ../../../libraries/ClusterOvsdb.robot
Resource          ../../../libraries/ClusterKeywords.robot
Variables         ../../../variables/Variables.py

*** Test Cases ***
Create Original Cluster List
    [Documentation]    Create original cluster list.
    ${original_cluster_list}    ClusterKeywords.Create Controller Index List
    Set Suite Variable    ${original_cluster_list}
    Log    ${original_cluster_list}

Check Shards Status Before Fail
    [Documentation]    Check Status for all shards in Ovsdb application.
    ClusterOvsdb.Check Ovsdb Shards Status    ${original_cluster_list}

Start OVS Multiple Connections
    [Documentation]    Connect OVS to all cluster instances.
    ${ovsdb_uuid}    Ovsdb.Add Multiple Managers to OVS    ${TOOLS_SYSTEM_IP}    ${original_cluster_list}
    Set Suite Variable    ${ovsdb_uuid}
```

External testing in OpenDaylight

External CI connected to ODL Gerrit

Notifications from Gerrit...

… and the ability to vote in Gerrit

Using all this to build robust applications

Test upstream when possible

Contribute your test cases

Writing good tests

Test for behaviour, not implementation

```java
@SuppressWarnings("unchecked")
@Test
public void testExecute() throws Exception {
    ReadWriteTransaction transaction = mock(ReadWriteTransaction.class);
    updatedBridgeRows = new HashMap<>();
    UUID uuid = mock(UUID.class);
    Bridge bridge = mock(Bridge.class);
    updatedBridgeRows.put(uuid, bridge);
    MemberModifier.field(OvsdbControllerRemovedCommand.class, "updatedBridgeRows").set(ovsdbControllerRemovedCommand, updatedBridgeRows);
    PowerMockito.mockStatic(SouthboundMapper.class);
    when(SouthboundMapper.createInstanceIdentifier(any(OvsdbConnectionInstance.class), any(Bridge.class))).thenReturn(mock(InstanceIdentifier.class));
    MemberModifier.suppress(MemberMatcher.method(OvsdbControllerRemovedCommand.class, "deleteControllers", ReadWriteTransaction.class, List.class));
    MemberModifier.suppress(MemberMatcher.method(OvsdbControllerRemovedCommand.class, "controllerEntriesToRemove", InstanceIdentifier.class, Bridge.class));
    ovsdbControllerRemovedCommand.execute(transaction);
    PowerMockito.verifyPrivate(ovsdbControllerRemovedCommand).invoke("deleteControllers", any(ReadWriteTransaction.class), any(List.class));
    PowerMockito.verifyPrivate(ovsdbControllerRemovedCommand).invoke("controllerEntriesToRemove", any(ReadWriteTransaction.class), any(Bridge.class));
}
```

Test your code, not the infrastructure

Use the infrastructure

```java
@Test
public void testCreateOvsdbController() throws Exception {
    try (InputStream resourceAsStream = SouthboundMapperTest.class.getResourceAsStream("openvswitch_schema.json")) {
        ObjectMapper mapper = new ObjectMapper();
        JsonNode jsonNode = mapper.readTree(resourceAsStream);
        System.out.println("jsonNode = " + jsonNode.get("id"));

        DatabaseSchema dbSchema = DatabaseSchema.fromJson(OvsdbSchemaContants.databaseName, jsonNode.get("result"));

        String uuidString = "7da709ff-397f-4778-a0e8-994811272fdb";
        OvsdbBridgeAugmentation omn = new OvsdbBridgeAugmentationBuilder()
                .setControllerEntry(Collections.singletonList(new ControllerEntryBuilder()
                        .setControllerUuid(Uuid.getDefaultInstance(uuidString))
                        .setTarget(Uri.getDefaultInstance("uri"))
                        .build()))
                .build();

        Map<UUID, Controller> returnedControllers = SouthboundMapper.createOvsdbController(omn, dbSchema);
        assertEquals(1, returnedControllers.size());
        Controller returnedController = returnedControllers.values().iterator().next();
        assertEquals("uri", returnedController.getTargetColumn().getData());
    }
}
```

```java
@Test
public void testOnSessionInitiated() throws Exception {
    ProviderContext session = mock(ProviderContext.class);
    when(session.getSALService(DataBroker.class)).thenReturn(db);
    TransactionInvokerImpl transactionInvokerImpl = mock(TransactionInvokerImpl.class);
    PowerMockito.whenNew(TransactionInvokerImpl.class).withArguments(any(DataBroker.class)).thenReturn(transactionInvokerImp
    PowerMockito.whenNew(OvsdbConnectionManager.class).withArguments(any(DataBroker.class), any(TransactionInvoker.class), c
    PowerMockito.whenNew(OvsdbDataChangeListener.class).withArguments(any(DataBroker.class), any(OvsdbConnectionManager.clas

    when(entityOwnershipService.registerListener(anyString(), any(EntityOwnershipListener.class))).thenReturn(mock(EntityOwn
    when(entityOwnershipService.registerCandidate(any(Entity.class))).thenReturn(registration);
    EntityOwnershipState entityOwnershipState = mock(EntityOwnershipState.class);
    when(entityOwnershipService.getOwnershipState(any(Entity.class))).thenReturn(Optional.of(entityOwnershipState));

    southboundProvider.onSessionInitiated(session);

    verify(entityOwnershipService, times(2)).registerListener(anyString(), any(EntityOwnershipListener.class));
    verify(entityOwnershipService).registerCandidate(any(Entity.class));
}
```

```java
@Test
public void testOnSessionInitiated() throws CandidateAlreadyRegisteredException {
    ProviderContext session = mock(ProviderContext.class);
    when(session.getSALService(DataBroker.class)).thenReturn(getDataBroker());

    // Indicate that this is the owner
    when(entityOwnershipService.getOwnershipState(any(Entity.class))).thenReturn(
            Optional.of(new EntityOwnershipState(true, true)));

    try (SouthboundProvider southboundProvider = new SouthboundProvider(entityOwnershipService,
            mock(OvsdbConnection.class))) {
        // Initiate the session
        southboundProvider.onSessionInitiated(session);

        // Verify that at least one listener was registered
        verify(entityOwnershipService, atLeastOnce()).registerListener(anyString(),
                any(EntityOwnershipListener.class));

        // Verify that a candidate was registered
        verify(entityOwnershipService).registerCandidate(any(Entity.class));
    }
}
```

Useful pattern: given, when, then

Given a certain state
(all your test-specific setup code goes here)

When something happens
(invoke the API you want to test)

Then verify the effect
(assert and verify)

```java
@Test
public void testConnect() throws UnknownHostException, InterruptedException, ExecutionException {
    // Given ...

    // When we request a connection ...
    InetAddress inetAddress = InetAddress.getByName("127.0.0.1");
    int port = 6640;
    IpAddress ipAddress = SouthboundMapper.createIpAddress(inetAddress);
    PortNumber portNumber = new PortNumber(port);

    final ConnectionInfo connectionInfo = new ConnectionInfoBuilder()
            .setRemoteIp(ipAddress)
            .setRemotePort(portNumber)
            .build();
    final InstanceIdentifier<Node> iid = SouthboundUtils.createInstanceIdentifier(connectionInfo);
    WriteTransaction transaction = dataBroker.newWriteOnlyTransaction();
    transaction.put(LogicalDatastoreType.CONFIGURATION, iid, SouthboundUtils.createNode(connectionInfo),
            WriteTransaction.CREATE_MISSING_PARENTS);
    transaction.submit().get();

    // Then the listener tries to open a connection
    Mockito.verify(ovsdbConnection).connect(inetAddress, port);
}
```

If your API doesn't change, your tests shouldn't change

API change with no functionality change

↓

the “when” part of your test changes

Functionality change

↓

the “then” part of your test changes

Complex code is worth making testable

Maintain your tests as you maintain your code

Maintain your tests as you maintain your documentation

Thank you!

Stephen Kitt, Red Hat
<skitt@redhat.com>