

Open Telekom Cloud

**Test Plan**

November 20, 2017

Sachin Jagtap

Email: Sachin.jagtap@click2cloud.net

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Name | Description of Change |
| 1.0 | 2016.11.21 | Sachin Jagtap | Initial document creation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version Approved | Role | Date |
| Sapan vaswani |  | Team Lead | 2016.11.20 |
| Sandeep Thakre |  | Project Manager | 2016.11.20 |
|  |  |  |  |
|  |  |  |  |

# Purpose

The purpose of this document is to outline the test strategy and overall test approach for the **OTC-VPC** project. This includes test methodologies, traceability, resources required, and estimated schedule.

# Audience

The audience of this document is the project team and the project management team. This document is also written for the extended test team. The test lead, testers, and any outsourced testers should be able to utilize this document to understand the scope of work that must be accomplished by the test team. The document is intended to accomplish its purpose only for the intended audiences.

# Introduction

The following test plan describes the formal testing to be performed by **Click2Cloud** QA team for **OTC- VPC** project. This test plan covers the included items in the test project, the specific risks to product quality we intend to address, the test environment, problems that could threaten the success of testing, test tools and harnesses we will need to develop, and the test execution process. Development of unit testing occurs outside of the test team’s area but QA team shall provide correct test environment so that they can be run in QA lab effectively as part of the testing process. This document also lays out the strategies, resources, and roles involved in performing **OTC- VPC** testing as a distinct testing subproject within the **Click2Cloud** QE.

Expansion of test coverage shall be a continuous process. The Test Team will develop manual tests to cover the quality risks identified after discussion with development team, as well as augmenting that test set to the extent possible under resource and time constraints with community based tests. However, when a release date closes by, **Click2Cloud** QA team will receive the tagged code base from development team approximately one week before the scheduled release date and will execute those tests against the tagged code base.

# Relationship to other documents

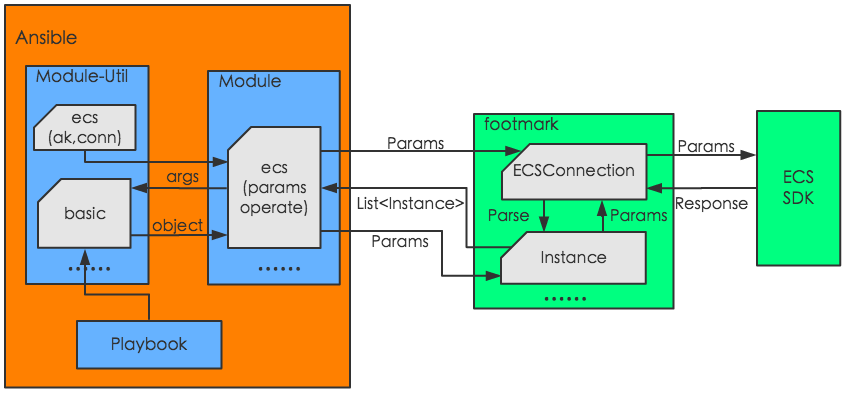
## Relationships to documents

Black box tests relating to use cases are developed from the use case diagram(s) in the RAD (requirements analysis document).

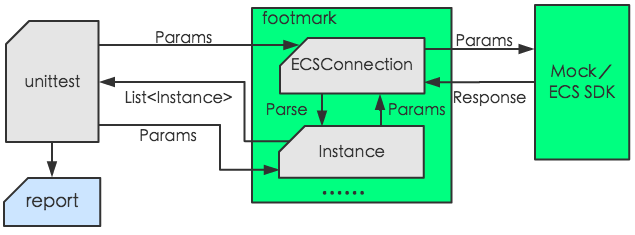
Black box tests derived from functional requirements are developed from the requirements lists in the RAD.

# System overview

## Software Architecture Overview



## Subsystem Decomposition (Components)



# Scope

## Features to be tested

1. Components developed in house
2. Testing on cloud environments
3. Components outsourced to be developed specifically for this project where this test team has primary responsibility to test and validate those components will be tested.
4. The table below defines the list of task that are covered in this project.

|  |  |
| --- | --- |
| **#** | **Interfaces** |
| **Instance Related Interface** | |
| 1 | Creating an instance |
| 2 | Modifying Instance Attributes |
| 3 | Querying Instance Status |
| 4 | Adding an Instance to a Security Group |
| 5 | Removing an Instance from a Security Group |
| **Disk Related Interfaces** | |
| 6 | Creating a Disk |
| 7 | Attaching a Disk |
| 8 | Detaching a Disk |
| 9 | Deleting a Disk |
| **Image Related Interfaces** | |
| 10 | Creating a User-defined Image |
| 11 | Deleting a User-defined Image |

## Features that will not be tested

1. 3rd party and Off-The-Shelf components [Aliyun Python SDK]
2. Infrastructure components [Aliyun Cloud Infrastructure]
3. The internet/WIFI backbone will be utilized during testing, however, no tests will be written executed to directly test the communications backbone.
4. Unit tests
5. Following table shows the task for which not a part of our development.

|  |  |
| --- | --- |
| **#** | **Interfaces** |
| **Instance Related Interface** | |
| 1 | Start-Stop an instance |
| 2 | Terminate an Instances |
| 3 | Get Instances |
| 4 | Get Instance information |

# Pass/Fail criteria

## Test Pass/Fail Criteria

The feature will pass or fail depending upon the results of testing actions. If the actual output from an action is equal to the expected output specified by a test case, then the action passes. Should any action within a test case fail the entire feature or sub-feature fails.

If a test case fails, it is not assumed that the code is defective. A failure can only be interpreted as a difference between expected results, which is derived from project documentation and actual results. There is always the possibility that expected results can be in error because of a misinterpretation of project documentation, incomplete documentation, or inaccurate documentation. It is also possible that a configuration error has occurred in the installation of one of the integrated applications.

## Pass/Fail Criteria:

Actual results equal expected results.

If a test case fails and a bug or issue is logged, communication of this action to the project team, through bug triage, is necessary to keep the group informed of progress.

# Suspension and resumption

## Suspension if:

functional tests fail

Issues in Aliyun Cloud Connectivity

does not deploy in priority I test environment

## Regression Testing

On a build by build basis, major bug fixes or code changes will be reviewed to determine the effects they may have on the system. If the changes are deemed to cause a sufficient amount of risk, regression test sets of the appropriately judged size will be created and executed.

A system-wide regression will also be run on the release candidate build to ensure incremental changes to the system have not altered the results of the tests that were run early in the test cycle.

## System Design Changes

If at any point in time issues are submitted that require a design change to the system, all testing will be suspended. After the changes to the requirements, system architecture, and object design are made, a review and updates will be performed of the test specifications to ensure they properly align with the revised system changes. After updates are made, testing will resume. Tests in the vicinity of the change must all be rerun. A 20% regression of other tests must also be performed to ensure the changes did not adversely affect other parts of the system.

# Testing materials (hardware/software requirements)

## Ansible Server Machine [Linux with GUI + Eclipse]

To enable the team to testing in an optimal environment, team needs an individual Ansible server machine with GUI and Eclipse installed.

Also, if a single system exhibits a strange bug, it can be left in that state for developer debugging and analysis, while testing continues on the other systems.

## Software required

1. **Eclipse IDE with YML Editor**
2. **Test Case Tracking**

**Test Manager** shall be used as test case management and tracking

1. **Bug Tracking**

**TFS** shall be used as bug tracking tool. Once a bug is resolved, a snippet of code and/or explanation shall be commented in **TFS** to better help understand the problem so that a test case can be added.

1. **Operating Systems**

* Linux

## Special requirements

Additional tools and software may need to be purchased or otherwise acquired or reused. Such software is used to execute special tests that are best run and results recorded and analysed by automated means.

# Test cases

## Test specifications derived from the use cases in the functional requirements

uc\_0.0.0.1\_ create\_ECSInstanceWithMandatoryParametrer

uc\_0.0.0.1\_ create\_ECSInstanceWithOptionalParametrer

## Test specifications derived from the system architecture specification [If any]

arch\_0.0.0.1\_ create\_ECSInstanceWithVPC

## Test case specifications that cover end-to-end test scenarios derived from the use cases

e2e\_\_ create\_ECSInstanceWithVPC

# Testing schedule

## Milestones

**Phase 1- Start Date: 12/26/2016**

|  |  |  |
| --- | --- | --- |
| Task Name | TIME | OWNER |
| Test Planning |  |  |
| Review Requirements documents | 4 Days | Test Manager |
| Create initial test estimates |  |  |
| Staff and train new test resources | 4 Days | Test Lead |
| First deploy to QA test environment | 1 Day | Test Lead |
| Functional testing with YML Files Creation– Iteration 1 | 2 Weeks | Tester |
| Iteration 2 deploy to QA test environment | 1 Day | Test Lead |
| Functional testing with YML Files Creation– Iteration 2 | 2 Weeks | Tester |
| System testing | 3 Days | Tester |
| Regression testing | 3 Days |  |
| UAT | 3 Days | Tester |
| Resolution of final defects and final build testing | 5 days | Tester |
| Deploy to Staging environment | 1 Day | Test Lead |
| Performance testing | 3 Days | Test Lead |
| Release to Production | 2 Days | Test Lead |

## Test Deliverables

|  |  |  |
| --- | --- | --- |
| Deliverable Name | Author | Reviewer |
| Test Plan | Test Lead | Project Manager |
| Functional Test Cases | Test Team | Test Lead |
| Logging Defects in TFS | Test Team | Test Lead |
| Daily/weekly status report | Test Team/ Test Lead | Test Lead/ Project Manager |
| Test Closure report | Test Lead | Project Manager |
|  |  |  |
|  |  |  |

## Responsibilities

**The Test Manager** is responsible for the overall test plan (this document) and test resources throughout the course of the project. He needs to be assigned to the project to review the requirements analysis, system architecture design, and object design of the system. From those specifications, he will generate the Test Plan. He will also generate section 9 “Test Cases” in this document. He will generate the list of test specifications and a brief description of each one in this document. He will generate and communicate the test strategy for the project to the test team and the rest of the project team as well as locate, acquire, and/or allocate the proper resources. He will provide periodic updates to the Program Director on the progress of test execution versus the plan as well as the metrics on the quality status of the product, focusing on key issues that need immediate attention from the Project Office.

**The Test Leads** are responsible for the creation of the detailed test specifications and will generate those and revise section 9 of this document as needed. The leads manage the day-to-day progress of each of their subcomponents and compile and report the metrics to the test manager. They are also responsible for ensuring the testers make adequate progress and follow the overall strategy defined by the Test Manager.

**The Functional Testers** are responsible for the test execution on a daily basis for the component of the system to which they’ve been assigned. They also lead the effort during most of the integration test cycle and hand off the testing to the System Testers during the last states of integration testing.

## Staffing and training needs

1 Test Manager

- responsible for training the following test resources

2 Test Leads

- responsible for process being used for this project

3 Functional Testers

- must be skilled with Ansible Technology

- must be skilled with YML file creation

- must be trained on python language

- must be skilled on the process being used for this project

- must be trained on the defect/issue tracking system utilized

## Risks and contingencies matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Probability** | **Risk Type** | **Owner** | **Contingencies / Mitigation Approach** |
| **SCHEDULE**  Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date. | High | High | Test Manager | * The testing team can control the preparation tasks (in advance) and the early communication with involved parties. * Some buffer has been added to the schedule for contingencies, although not as much as best practices advise. |
| **HARDWARE**  Unable to acquire some of the necessary hardware and software required for integration and system testing |  |  | Program Manager  Test Manager  Development Manager | * Utilize existing acquired hardware. * Split test execution into morning and evening shifts such that testing can occur for multiple teams in the same day using the limited hardware. This requires support of the development during both shifts. |
| **CLOUD CONNECTIVITY**  Delayed due to the Network connectivity or slow response from Cloud | Medium | High | Program Manager  Test Manager  Development Manager | \_\_\_\_\_\_\_\_\_\_\_\_ |
| **DEFECTS**  Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve. | Medium | High | Testing Team | * Defect management plan is in place to ensure prompt communication and fixing of issues. |
| **SCOPE**  Scope completely defined | Medium | Medium | Test Lead | * Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing. |
| Delayed Testing Due to new Issues | Medium | High | Testing Team | * During testing, there is a good chance that some “new” defects may be identified and may become an issue that will take time to resolve. * There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved. * If these issues become showstoppers, it will greatly impact on the overall project schedule. * If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Disclaimer

This DRAFT Specification is being forwarded to you strictly for informational purposes and sign-off requirement before development starts. This document covers functional and technical requirement of modules and interfaces implementation with Ansible for Aliyun (Alibaba) Cloud. The specification is "AS IS," "WITH ALL FAULTS" and Click2Cloud makes no warranties, and disclaims all warranties, express, implied, or statutory related to the specifications. THE CORPORATIONS ARE NOT LIABLE FOR ANY INCOMPLETENESS OR INACCURACIES. THE CORPORATIONS ARE NOT LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES RELATING TO THE SPECIFICATIONS OR THEIR USE.

# Appendix A: Glossary

|  |  |  |
| --- | --- | --- |
| No. | Initial Name | Description |
| 1 | CR | Change Request |
| 2 | FSD | Functional Specification Document |
| 3 | ECS | Elastic Computing Service |