DevTest 8 – Best Practice Architecture

Overview

# Introduction

Customers often ask “Where do we Start” when setting up and configuring their DevTest architecture. Of course, this really depends on how they are want to use DevTest - whether using Virtual Services or running tests or maybe a combination of both. So based on potential usage of DevTest, we started building a Best Practice Architecture Guide based on real world deployments.  As a part of this project, we will build out several architectures that are implemented by either customers or CA Services or are new/tested architectures that fit a specific need.

Our intention is to provide architecture options for customers, partners and CA Services that can be used as reference material for designing or verifying implementation plans.

# Architecture Examples

All DevTest server installations are distributed. All DevTest installations have a SCM repository server attached to store project specific artifacts (test cases, virtual services) and their versions.

* Hybrid environment (on premise and public cloud) – server components are installed in the cloud, except for the Enterprise Dashboard service.
* Everything Cloud (public cloud) – all server components are installed in the cloud
* Multiple Teams – multiple DevTest server installations in the cloud
* Large Team – single DevTest server installation with multiple VSEs and Simulators for load balancing
* Shared DevTest Infrastructure – DevTest server installation for use by internal and external clients

Contents

[Introduction 1](#_Toc433371756)

[Architecture Samples 1](#_Toc433371757)

[#1 Hybrid environment 2](#_Toc433371758)

[#2 Everything Cloud 2](#_Toc433371759)

[#3 Multiple Teams 3](#_Toc433371760)

[#4 Large Team 3](#_Toc433371761)

[#5 Shared DevTest Infrastructure 4](#_Toc433371762)

# #1 Hybrid environment

Customer wants to deploy some or all DevTest components in public cloud.



The lists of available port numbers in above diagram are incomplete.

# #2 Everything Cloud

Customer wants to deploy all DevTest server components in public cloud.



The lists of available port numbers in above diagram are incomplete.

# #3 Multiple Teams

Customers want to share DevTest infrastructure among teams. Each team gets a separate environment. A team can be an external partner.

A separate DevTest environment requires a separate registry. Multiple registry can report usage metrics to a single Enterprise Dashboard service. This architecture deploys a central Enterprise Dashboard server, which communicates to all the separate Registry services. Essentially, this architecture is a multiple deployment of the cloud components of the Hybrid architecture sample. In the diagram below each client aligns to its Azure environment, i.e. Client-1 to Azure-1, Client-2 to Azure-2, etc.



The lists of available port numbers in above diagram are incomplete.

# #4 Large Team

Customer wants to deploy a distributed DevTest environment for a large team that shares resources.

This architecture uses separate VMs per VSE or Simulator. Because all register with the same Registry service VSEs and Simulators must get distinct names assigned in post installation to show up in Portal, Workstation and Server Console by different names. As there is only one VSE or Simulator configured on each system different names can be configured in local.properties file of the local DevTest Server installation.

Alternatively, multiple VSEs or multiple Simulators can be installed on a single system. Then, on Windows systems, they should not be installed as services. A start script needs to be created to start the multiple processes. Again, VSEs and Simulators must get distinct names assigned. Different names are assigned to processes during process start. In addition, each VSE and each Simulator must be configured to listen on different ports.

Also, container based installations (Docker) of VSEs and Simulators are possible. Container based installation also requires diligent naming and network configuration.



The lists of available port numbers in above diagram are incomplete.

# #5 Shared DevTest Infrastructure

Customers want to allow external partners to use their internal DevTest infrastructure for tests. Example: Customer has built a validation test suite that a 3rd party software must pass in order to be certified. Partners can run the validation tests against their SUT, and test results are shared between the customer and its partner. (See CA ITC team (DVNAT))



The lists of available port numbers in above diagram are incomplete.