DevTest 8.x – Best Practice Architecture

#1 - Hybrid environment (on premise and public cloud)

Introduction

Customers often ask “Where do we Start” when setting up and configuring their DevTest architecture. Of course, this really depends on how they plan 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 initiative, 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 options for customers, partners and CA Services on how to implement DevTest within their environment to meet the specific use cases.

This document will contain guidance on implementing DevTest on premise, in private or public clouds or hybrid environments. It will describe the specific DevTest Capabilities.

This first edition walks through setting up DevTest 8.4 in a hybrid environment, meaning that some DevTest components are installed on premise (on physical systems or in private cloud) and others are installed in public cloud, namely in Microsoft ® Windows Azure (Azure).

Document Changes

Version	Date	Primary Author	Description
1.0	11/06/2015	Ulrich Vogt Koustubh Warty	Initial creation #1 - Hybrid environment (on premise and public cloud)

Contents

Business Case

This architecture sample covers the business case of a customer who wants to deploy a distributed DevTest configuration in a public cloud for internal or external partners to access.

In this sample, implementation of this DevTest architecture the Enterprise Dashboard component is installed in the DMZ. All the other DevTest components are installed in Microsoft Azure environment on individual VMs as explained in the document.

Architectural Restrictions

Following restrictions apply to a DevTest architecture and are recommended to follow:

1. To warrant performance Registry database must be located electronically close to DevTest server components. Please see Database Requirements for details.
2. Enterprise Dashboard database contains audit data. Resources related to audit data must not be released.

To meet these requirements the following architecture design decisions have been made:

1. Enterprise Dashboard database is located on premise
2. Registry database is located on the same network as the Registry service

This architecture is designed for Service Virtualization and Application Test.

Architecture Diagram

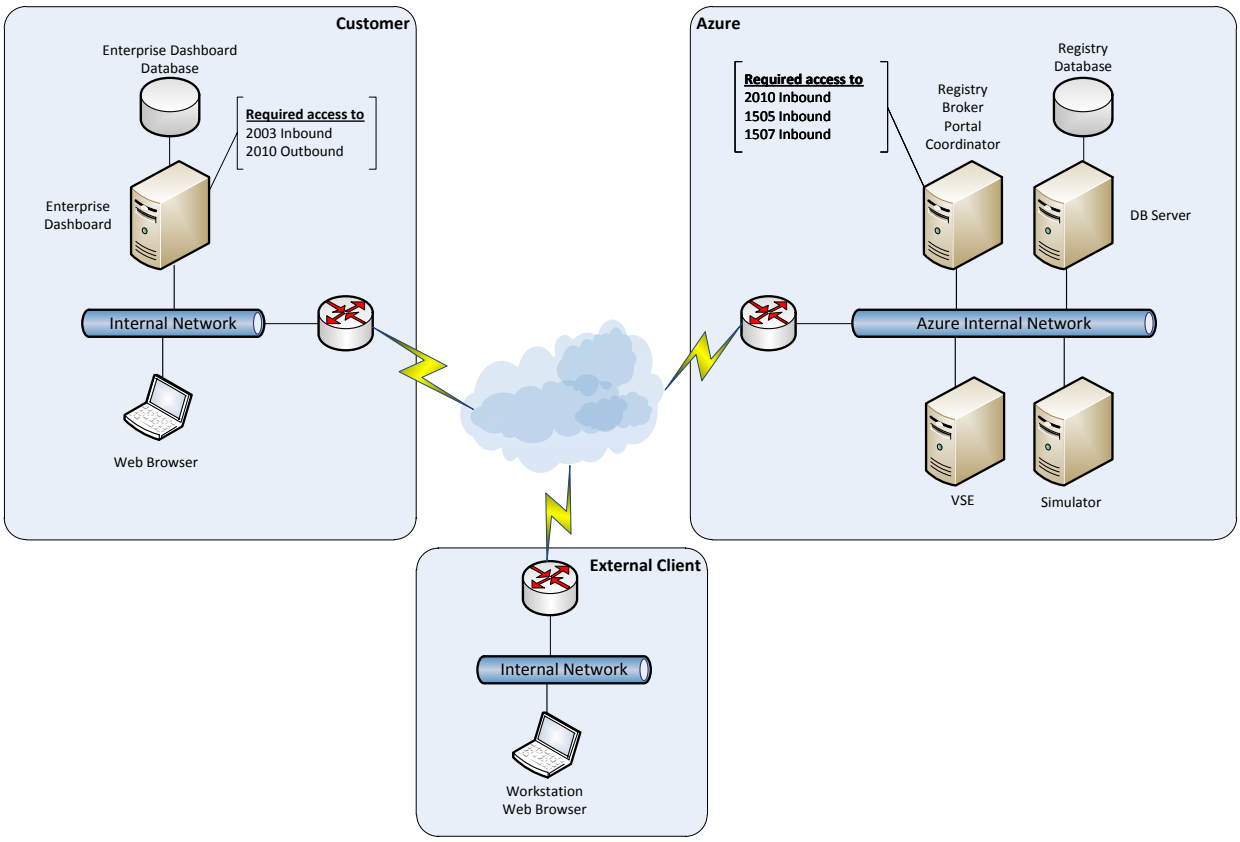


Figure 1 - Architecture Diagram

The Architecture diagram shows ports that need to be open for external DevTest service access. For the sake of simplicity, it does not show ports that needs to be open for other services, such as Remote Desktop Protocol or Windows PowerShell.

Implementation Details

Installation of some DevTest components depends on business case requirement. Most components are required regardless of use case to provide the DevTest infrastructure.

DevTest Server Components

In order to run Service Virtualization or Application Test in cloud environments DevTest VSE and DevTest Simulator service must be provided. These components require additional DevTest services installed and running:

DevTest service	**Requires**
Simulator	Coordinator, Registry
VSE	Registry
Shared among DevTest components	Portal

- Enterprise Dashoard

  The Enterprise Dashboard service and its database are installed on customer's premise to make sure that audit data are available for compliance checks. For the sake of simplicity, the database is installed locally on the same system as the Enterprise Dashboard service, but a dedicated database server can also be used.

  Installation of the Enterprise Dashboard service is mandatory.

- Registry

  Registry service and its database are installed in the cloud. The Registry database is installed on a dedicated database server to better visualize the need for a database resource. The Registry Service requires access to the Enterprise Dashboard service for license information and for providing usage data.

  Installation of a Registry service is required for every DevTest use case.

- Broker

  Broker Service is installed on the same system as the Registry Service. Broker Service is required for Continuous Application Insight.

- Portal

  Portal Service is installed on the same system as the Registry Service. Portal Service is required for Web access to DevTest server components

- Coordinator

  Coordinator Service is installed on the same system as the Registry Service. Coordinator Service is required to stage test cases to Simulator Services. A Coordinator Service can manage multiple Simulator services.

  Coordinator services are required for Application Test use cases only.

- Simulator

  Simulator service is installed on a separate system. If multiple Simulator services are required for scalability reasons, they are assumed to be installed on different systems.

  Simulator services are required for Application Test use cases only. Therefore VMs for Simulator services can be provided on demand.

- Virtual Service Environment (VSE)

  VSE service is installed on a separate system. If multiple VSE services are required for scalability reasons, they are assumed to be installed on different systems.

  VSE services are required for Service Virtualization only. Therefore VMs for VSE services can be provided on demand.

Network

The Azure network blocks any inbound network traffic that is not required to access the provided systems. Any ports that are needed to access DevTest components must be published in Azure. Some well-known port numbers in DevTest might not be available for publishing, but must be mapped or configured to different port numbers.

Similarly, customer and client networks might also be restricted for inbound and outbound traffic. These firewalls also have to be configured to allow access to the DevTest ports in Azure.

System Setup

DevTest is set up in a distributed configuration. It is distributed in two separate environments, in the DMZ and in Microsoft Azure.

- DMZ

Service	Host name	Public IP
Enterprise Dashboard	devtest-edb	207.138.132.69

- Microsoft Azure

Service	Host name	Public IP
Registry, Broker, Coordinator, Portal	devtest-regcoor	40.122.206.141:57579
Simulator	devtest-sim	40.122.206.141:57875
VSE	devtest-vse	40.122.206.141:63306
SQL Server	devtest-sql	40.122.206.141:65446

Note: The ports in the table above 57579, 57875, 63306 and 65446 are all ports that are used to connect to the VM using RDP.

The Enterprise Dashboard server runs in DMZ. It is configured with following resources:

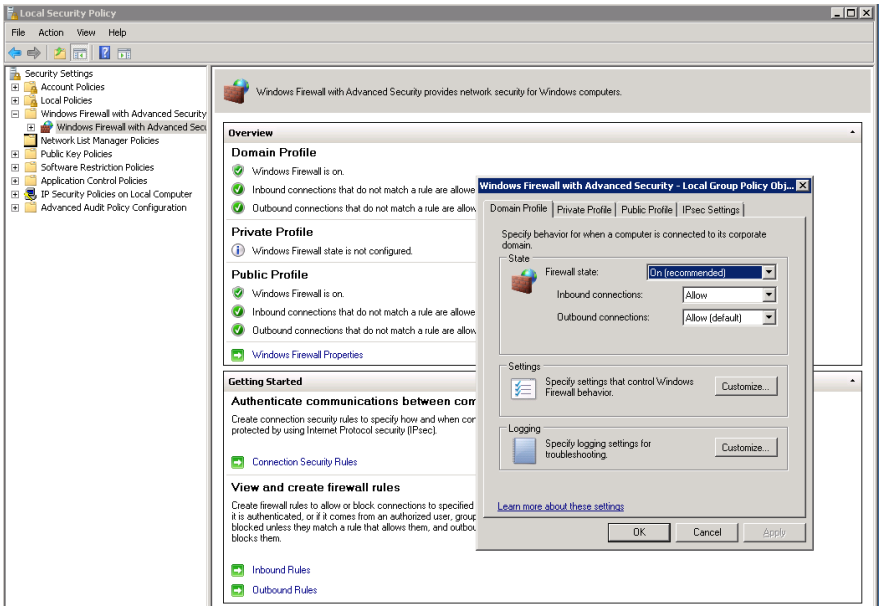
Host name	CPU (Cores)	Memory	#Disks / Disk size	#NICs
devtest-edb	2 Cores	3.5GB	1 x 60GB	1 NIC

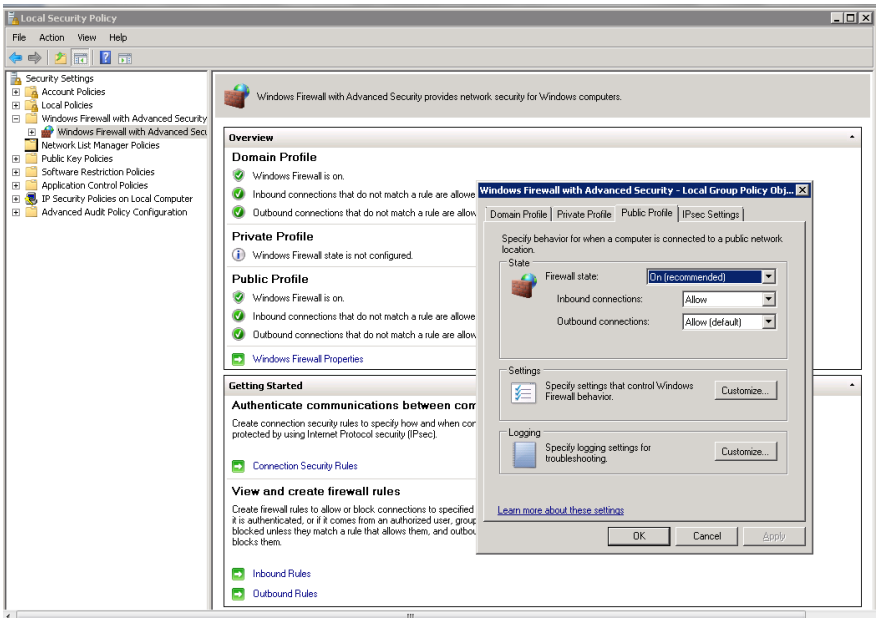
Following ports were opened: 1506 and 2003 both bidirectional.

Firewall Policies

All virtual machines of the DevTest cloud in Azure are configured with identical firewall settings:

- Same firewall settings for Domain Profile, Private Profile and Public Profile
    - Firewall is activated (on) for Domain and Public Profiles, but off for Private Profile
- If this firewall is not configured accordingly then there are no Inbound and Outbound connections available between the VMs and the outside world





Azure Setup

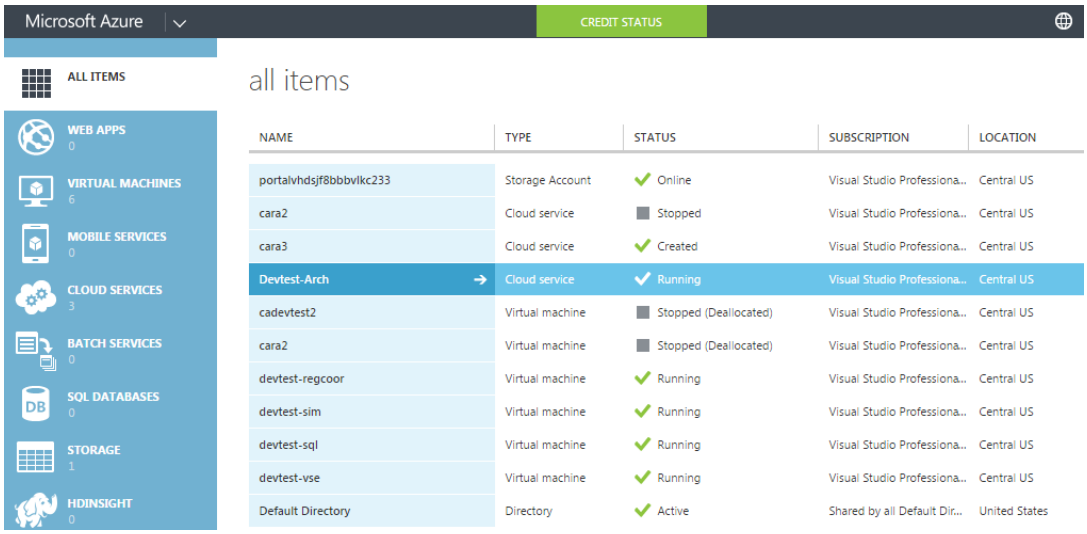
The Microsoft Azure setup of the DevTest components requires to create and to configure

- Cloud Service to configure. A cloud service is required before creating any VM on the Azure site. The cloud service hosts all the VMs, databases, webapps, etc. If a cloud service does not exist then the administrator is prompted to create a new one during the provisioning of the first VM.
- The set of Virtual Machines to run the assigned DevTest components

The virtual machines are all based on Microsoft Windows 2012 Server, which is a 64-Bit system.

Cloud Service

In Azure, the Cloud Service 'DevTest-Arch' defines the environment for the virtual machines discussed in this document.

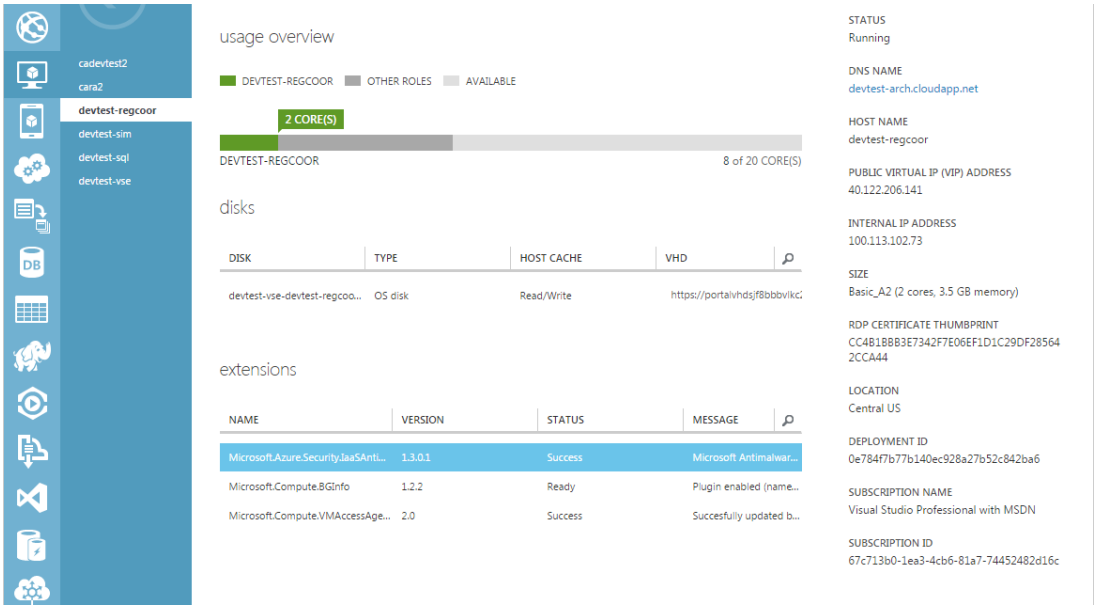


Virtual Machines

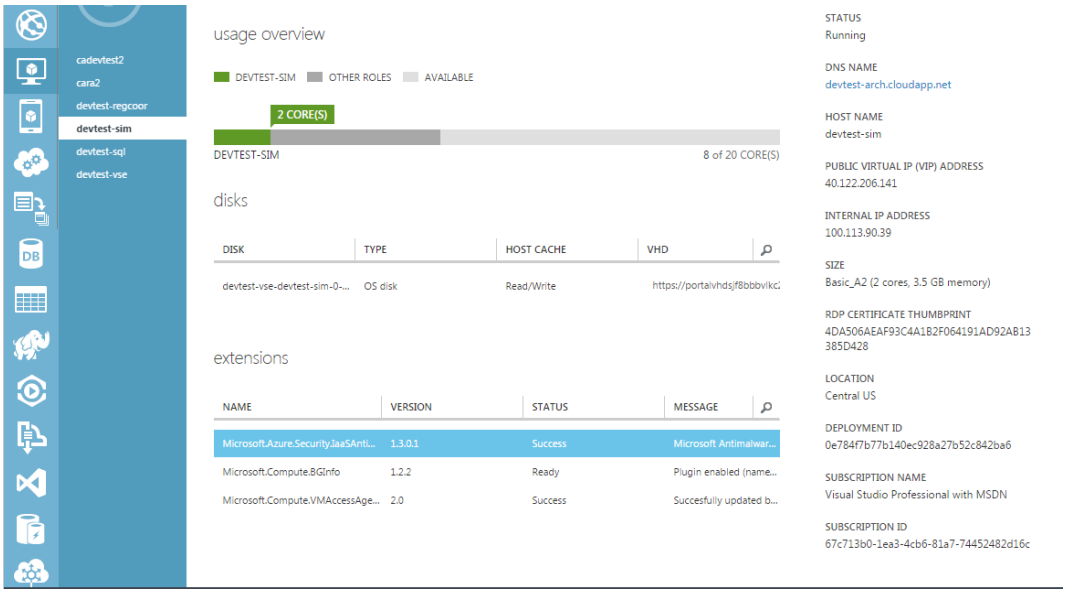
The other DevTest Server components run in Microsoft Azure. They are configured with following resources:

Host name	Public IP	Private IP	CPU (Cores)	RAM	#Disks	#NICs
devtest-regcoor	40.122.206.141	100.113.102.73	2 Cores	7.0 GB	1 x 60GB	1 NIC
devtest-sim	40.122.206.141	100.113.90.39	2 Cores	3.5GB	1 x 60GB	1 NIC
devtest-vse	40.122.206.141	100.113.74.29	2 Cores	3.5GB	1 x 60GB	1 NIC
devtest-sql	40.122.206.141	100.113.116.56	2 Cores	3.5GB	1 x 60GB	1 NIC

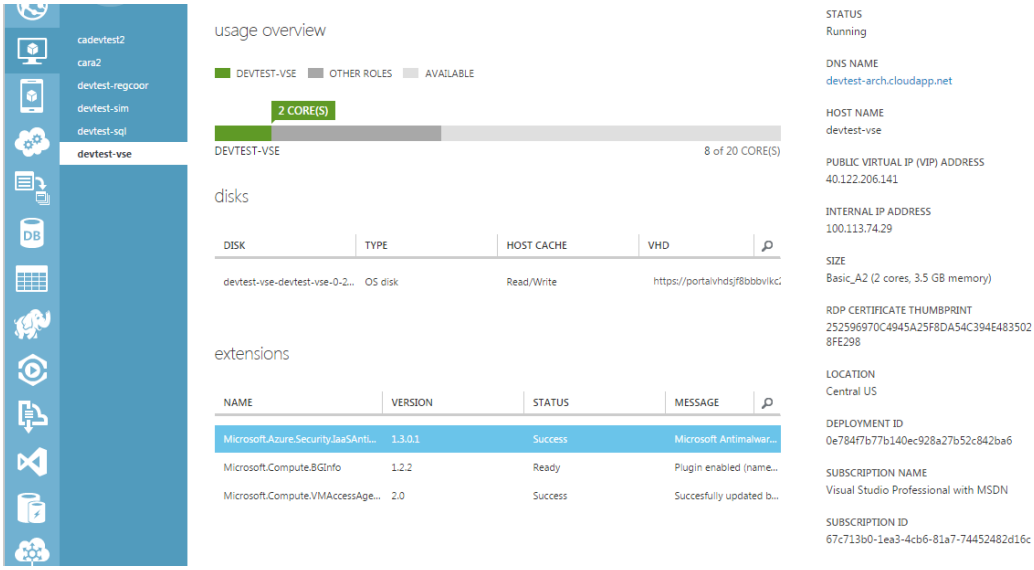
VM for Registry/Coordinator/Portal



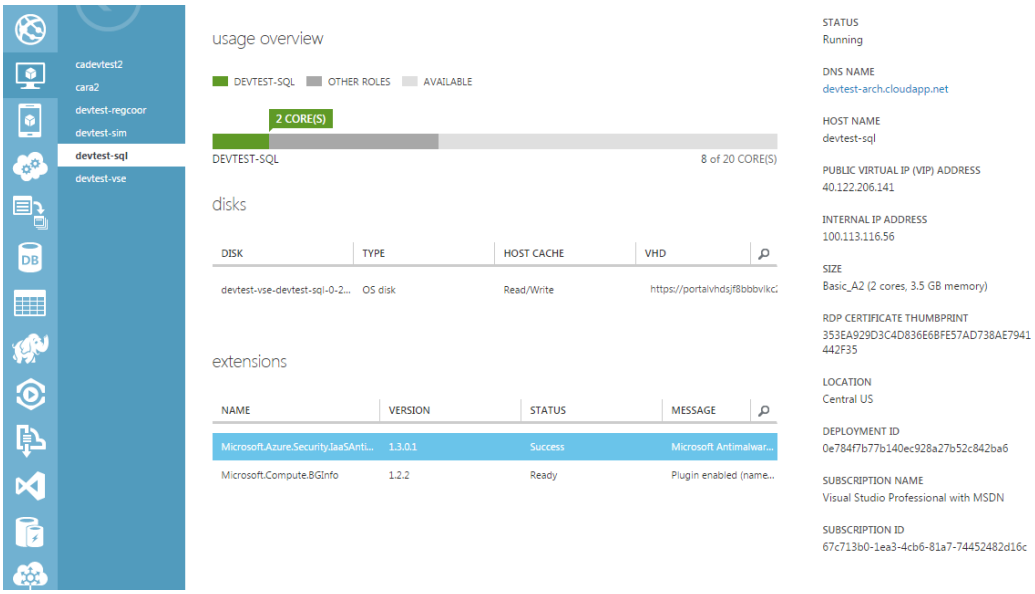
VM for Simulator



VM for VSE



VM for SQL Server



HTTP/TCP Ports exposed for each of the DevTest Components

Azure requires and offers a NAT configuration to expose access to services available on the virtual machines deployed in Azure.

External access to following virtual machines in Azure must be configured

Server	**VM name**
Registry/Coordinator/Portal	Devtest-regcoor
Simulator	Devtest-sim
VSE	Devtest-vse

Following services on these virtual machines running Microsoft Windows Server need to be accessed from outside

- Windows Services

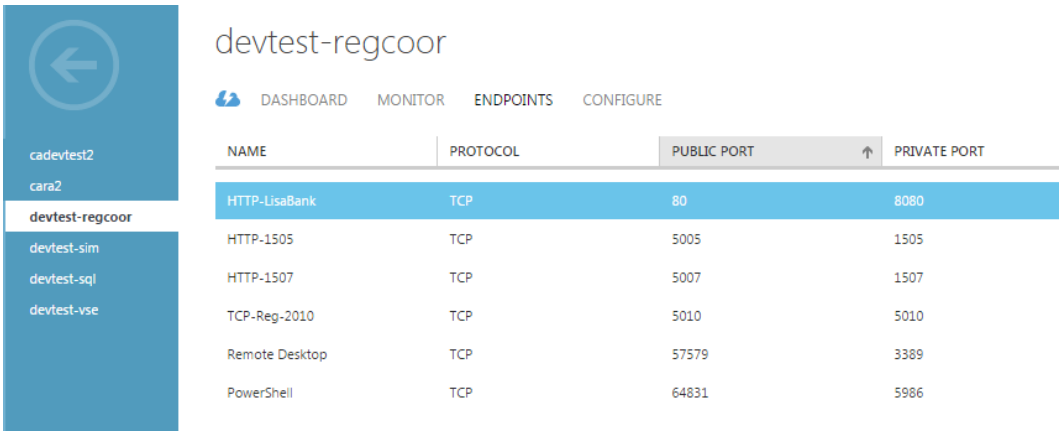
Service name	**Default Port no.**
Remote Desktop	3389
PowerShell	5986

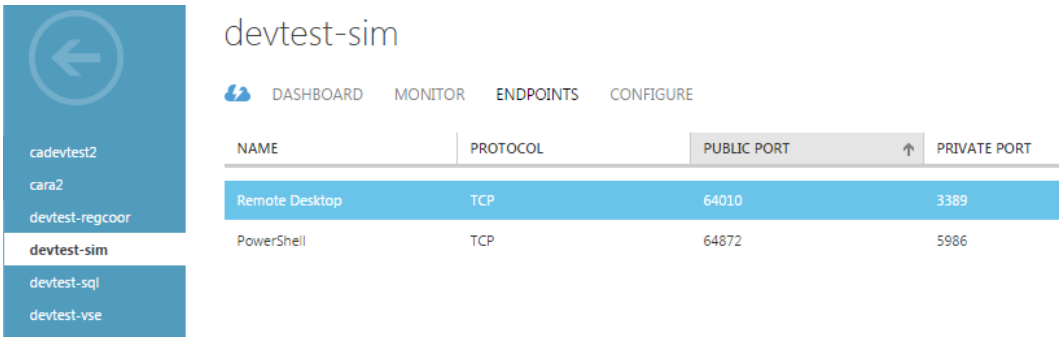
- DevTest Services

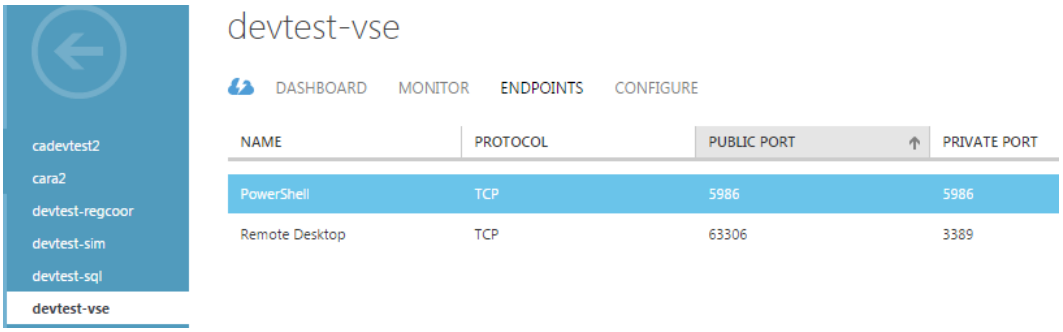
Service name	**Default Port no.**
Registry	2010
Portal	1507
Server Console	1505
Demo Server	8080

When DevTest components are installed on VMs in the cloud they need to access the DevTest Enterprise Dashboard service. Therefore, outbound access to the Enterprise Dashboard service from MS Azure VMs to Enterprise Dashboard service on customer premise on default port 2003 must be available.

The following screenshots show the public port settings and mappings to private ports on the different VMs







Setup Verification

This section covers steps to verify that the Azure setup is working correctly as expected. This includes steps to

- Determine that components are connected to the DevTest Registry service
- Verify that test cases can be launched from Workstation

- Verify that virtual services can be deployed from Workstation
- Check if tests can be monitored in DevTest Portal
- Check if virtual services show up in DevTest Server Console

These verification steps are executed from Workstation, Portal and Server Console to test the various access methods.

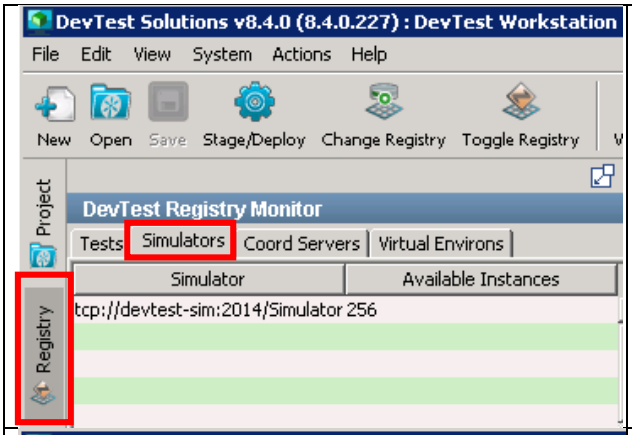
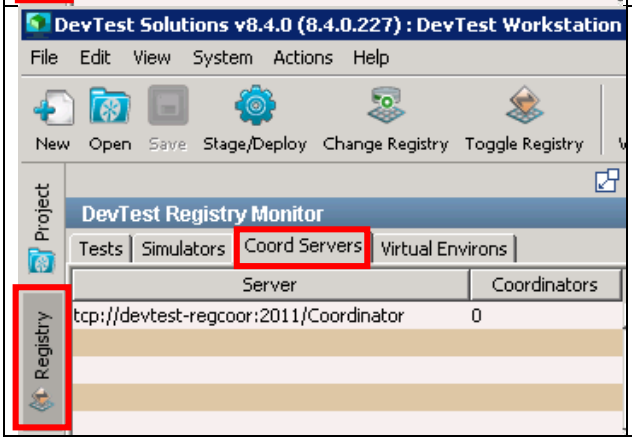
DevTest Workstation (from within the company network)

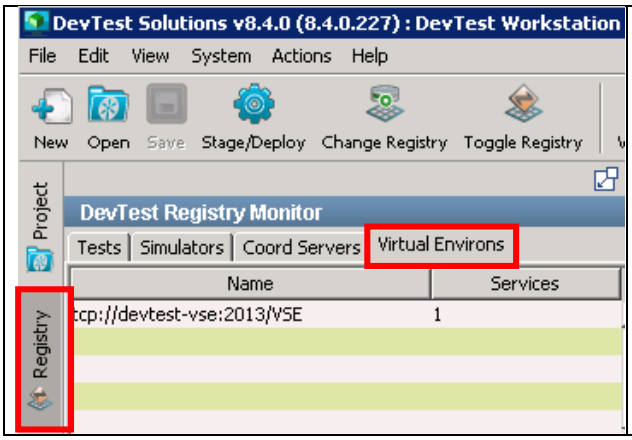
Workstation actions are used to access DevTest Server components:

- Connectivity to required DevTest Server components such as Simulator, Coordinator and VSEs
- Staging a test to validate access to Coordinator and Simulator
- Deployment of a virtual service to verify access to VSE

When launching Workstation access to registry is required for user authentication. Therefore, a successful start of Workstation already verifies a working connection to the registry service.

Component Connectivity Check

	To check that Simulator service is available • Select Registry tab • Click on Simulator tab and see the URL and the number of available Simulator instances
	To check that Coordinator service is available • Select Registry tab • Click on Coord Servers tab and see the URL and the number of available Coordinator instances

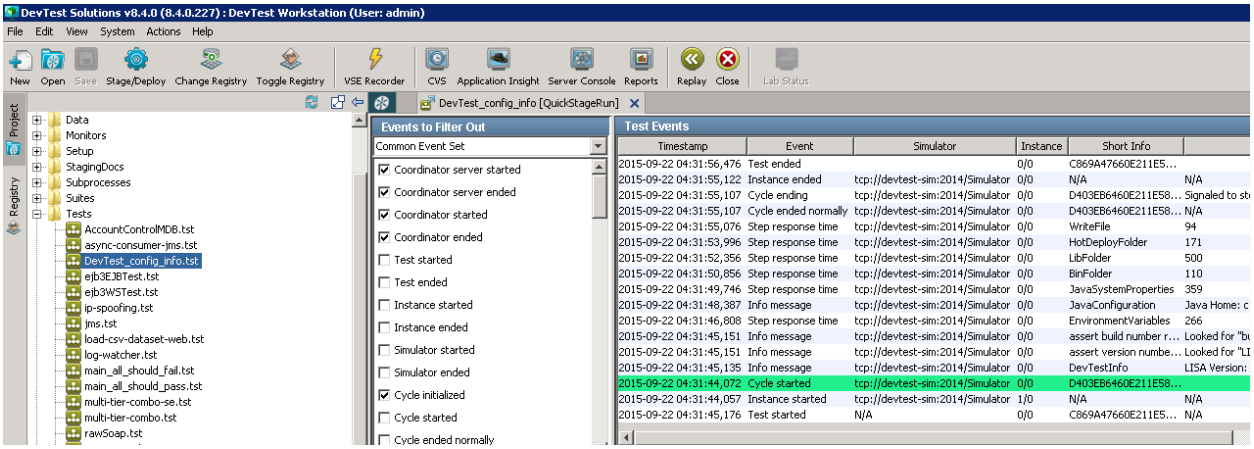


To check that VSE service is available

- Select Registry tab
- Click on Virtual Environs tab and see the URL and the number of available VSE instances

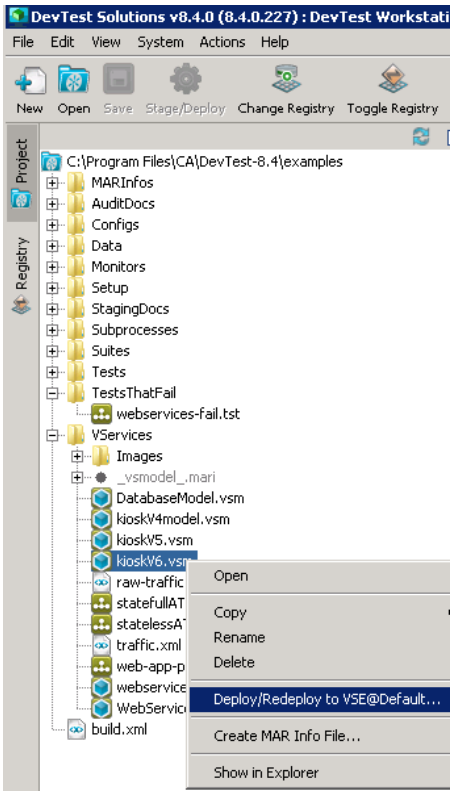
Running Tests from Workstation

Stage a sample test from Workstation and verify the test events for proper test execution.



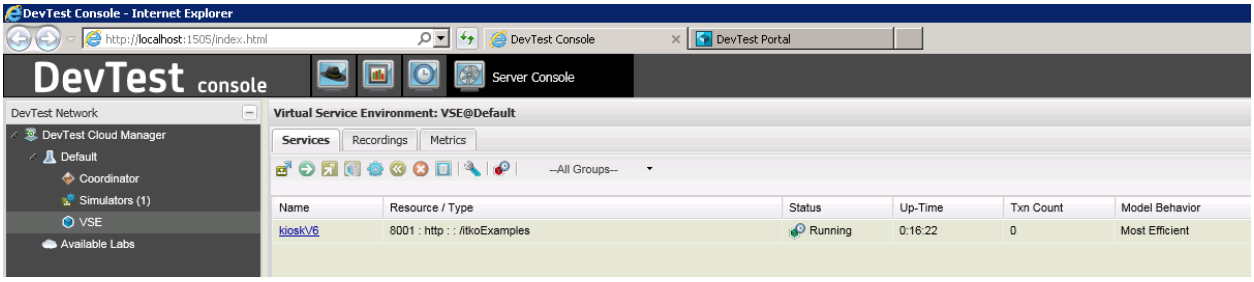
Deploy Virtual Service to VSE

Deploy a sample virtual service from examples project to verify that VSE can be accessed from Workstation.



VSE Console

In order to test access to DevTest Server Console, open VSE in Server Console to verify that the deployed VS is running.

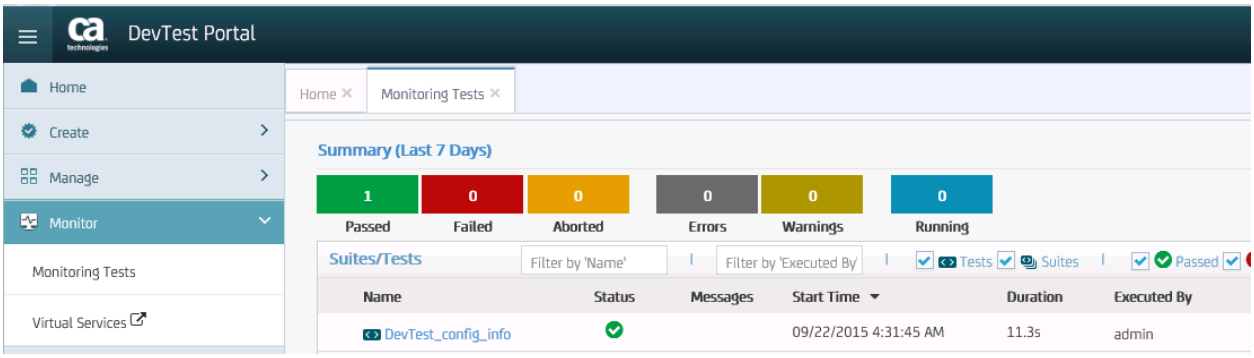


DevTest Portal

Access to DevTest Portal service can be tested by checking if the sample test that was staged from Workstation can be monitored.

Monitor Test

Open DevTest Portal, navigate to ‘Monitor / Monitoring Tests’ and verify the status of the previously staged sample test.


DevTest Portal
Home
Create
Manage
Monitor
Monitoring Tests
Virtual Services
Home
Monitoring Tests
Summary (Last 7 Days)
1
Passed
0
Failed
0
Aborted
0
Errors
0
Warnings
0
Running
Suites/Tests
Filter by 'Name'
Filter by 'Executed By'
Tests
Suites
Passed
Name
Status
Messages
Start Time
Duration
Executed By
DevTest_config_info
09/22/2015 4:31:45 AM
11.3s
admin