DevTest 9.x – Best Practice Architecture

#4 – Large Team – single DevTest server installation with multiple VSEs and Simulators for load balancing

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 customers or CA Services have implemented, or that 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 fourth use case walks through setting up DevTest 9.0 in a Microsoft® Windows Azure (Azure) environment for a large development team that needs multiple and separated resources to create, test and operate automated tests and virtual services. The Source Control Management (SCM) is located on premise (on physical systems or in private cloud).

Document Changes

Version	Date	Primary Author	Description
1.0	11/01/2016	Ulrich Vogt, Koustubh Warty	Initial creation

Contents

Business Case

This architecture sample covers the business case of a customer who wants to deploy a DevTest configuration in a public cloud for internal or external partners to access. Multiple developers who are building DevTest assets such as automated test cases, and virtual services will access this DevTest installation. As these assets are in different stages of development, best practices indicate they should not be deployed to the same simulators/VSEs.

In this sample, implementation of DevTest architecture consists of installing and configuring all the DevTest components in Microsoft® Azure environment on individual VMs as explained in the document. In particular, there will be several dedicated VSE and Simulator systems connecting to a single repository. We believe that a single VSE or Simulator instance per node is the simplest option for administering the Dev Test environment in the cloud. Of course, a single system can host multiple VSEs or Simulators, resp., on different ports. However, the cloud allows for easy addition of systems or extension of system resources to accommodate growth or organizational needs, which we think, will better serve the various purposes.

The SCM will be on the premise. Every time a DevTest user works on any DevTest assets namely Test Cases, Virtual Services, mar files, etc., they will checkout from the SCM server locally, and after modification of those assets, will check back the modified asset(s) to the SCM server.

Architectural Considerations

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.
3. The SCM used here is the free standard edition of the Visual SVN server from https://www.visualsvn.com/server/
4. Every user machine will have a client that can talk to the SVN server. In this case, we have used the free version of Tortoise SVN Client from https://tortoisesvn.net/downloads.html

   Note: This document does not go into the details on installing and configuring the SVN server and the client.

This architecture design applies to Service Virtualization and Application Test.

Note: In this architecture, the DB server is part of the cloud installation, and therefore, by default, supposed to be available for a limited amount of time only. Dropping the Virtual Machines in the cloud used in this DevTest architecture will also delete all DevTest reports stored in database. If the customer wants DevTest reports to be available after destroying the DevTest environment, s/he needs to consider a DB backup strategy.

Architecture Diagram

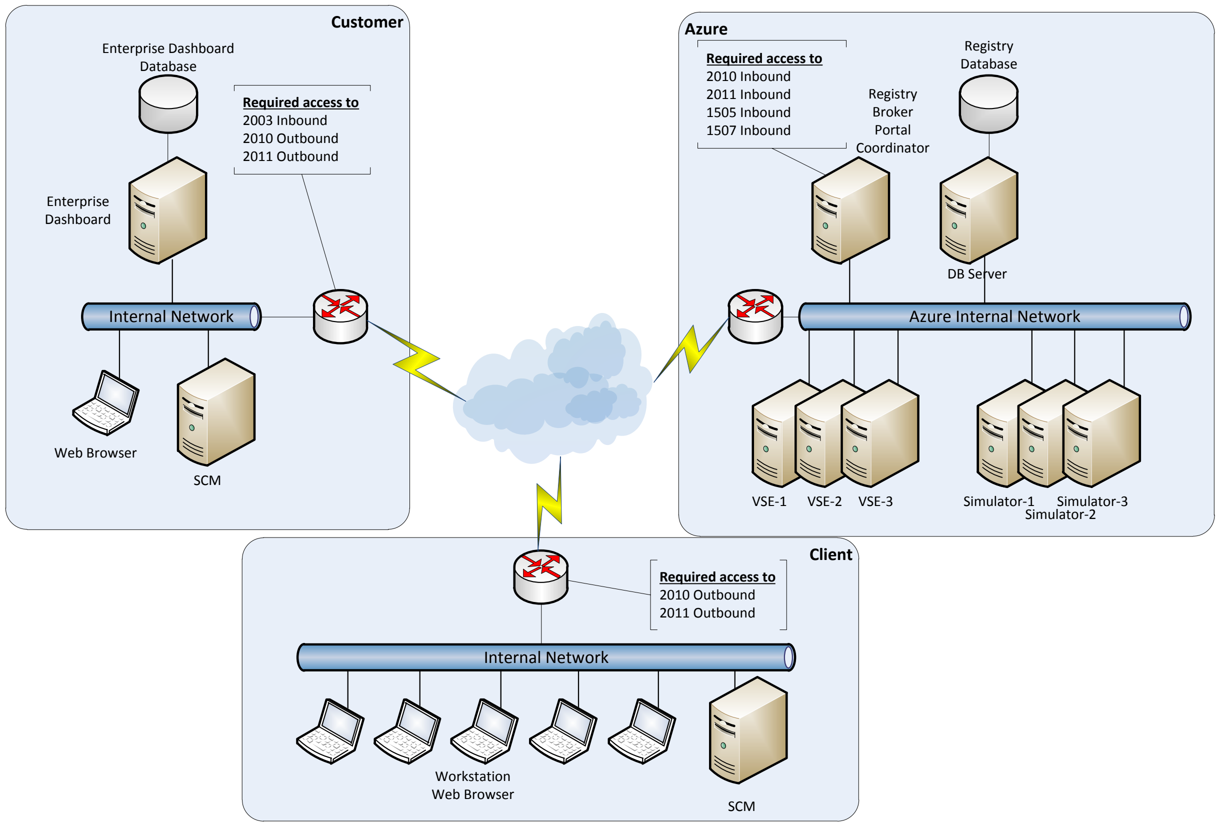


Figure 1 - Architecture Diagram

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 Dashboard

  The Enterprise Dashboard service and its database is installed on the Azure VM in the cloud. 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 as well. 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.

System Setup

DevTest is deployed in a distributed configuration. It is distributed in two separate environments, in DMZ and in Microsoft® Azure. DMZ hosts the local Workstation, DevTest Server runs in Microsoft® Azure.

Microsoft Azure Setup - Overview

Microsoft® offers two types of portals to manage a Microsoft® Azure environment. For setting up this architecture, we used the Azure portal under https://manage.windowsazure.com. At the time of creating this document, the new portal (https://portal.azure.com) did not seem to support adding a new VM to an existing cloud service.

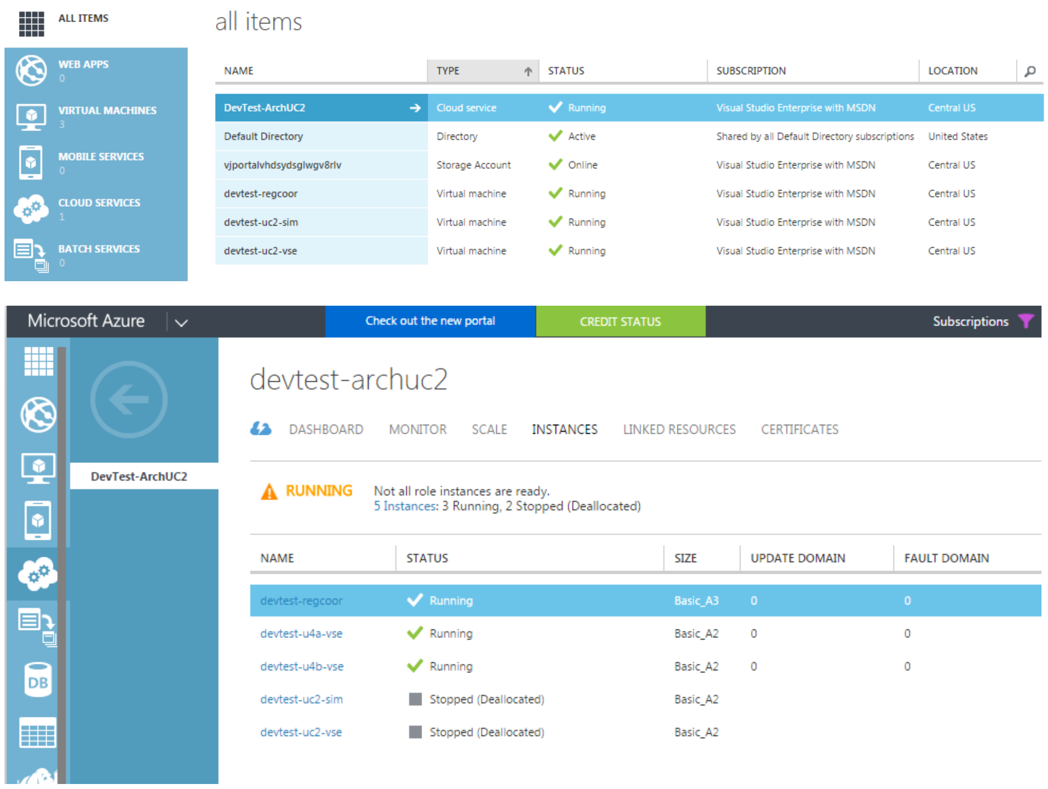
The DevTest components are created using the classic approach. The new Azure portal supports VM provisioning by a Resource Manager, which we did not use yet for this exercise.

The classic approach requires to create and to configure

- Cloud Service:
    - The cloud service hosts all the VMs, databases and webapps, etc.
    - The cloud service becomes the publicly addressable entity of the entire DevTest installation in Microsoft Azure. With current setup of the DevTest server environment Microsoft Azure assigns a public IP address to the cloud service dynamically when it is started.
    - Users access individual VMs by VM specific ports of the cloud service. For instance, each VM in this cloud service has a unique RDP port assigned in cloud service.
    - A cloud service is required before creating any VM on the Azure site. If a cloud service does not exist then Microsoft® Azure prompts the administrator 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 used for DevTest server installation are all based on Microsoft® Windows 2012 Server, which is a 64-Bit system.

Cloud Service

In Microsoft® Azure, the Cloud Service 'DevTest-ArchUC2' defines the environment for the virtual machines discussed in this document.



all items

NAME	TYPE	STATUS	SUBSCRIPTION	LOCATION
DevTest-ArchUC2	Cloud service	✓ Running	Visual Studio Enterprise with MSDN	Central US
Default Directory	Directory	✓ Active	Shared by all Default Directory subscriptions	United States
vjportalvhdsydsglwgv8rlv	Storage Account	✓ Online	Visual Studio Enterprise with MSDN	Central US
devtest-regcoor	Virtual machine	✓ Running	Visual Studio Enterprise with MSDN	Central US
devtest-uc2-sim	Virtual machine	✓ Running	Visual Studio Enterprise with MSDN	Central US
devtest-uc2-vse	Virtual machine	✓ Running	Visual Studio Enterprise with MSDN	Central US

devtest-archuc2

RUNNING Not all role instances are ready.
5 Instances: 3 Running, 2 Stopped (Deallocated)

NAME	STATUS	SIZE	UPDATE DOMAIN	FAULT DOMAIN
devtest-regcoor	✓ Running	Basic_A3	0	0
devtest-u4a-vse	✓ Running	Basic_A2	0	0
devtest-u4b-vse	✓ Running	Basic_A2	0	0
devtest-uc2-sim	■ Stopped (Deallocated)	Basic_A2		
devtest-uc2-vse	■ Stopped (Deallocated)	Basic_A2		

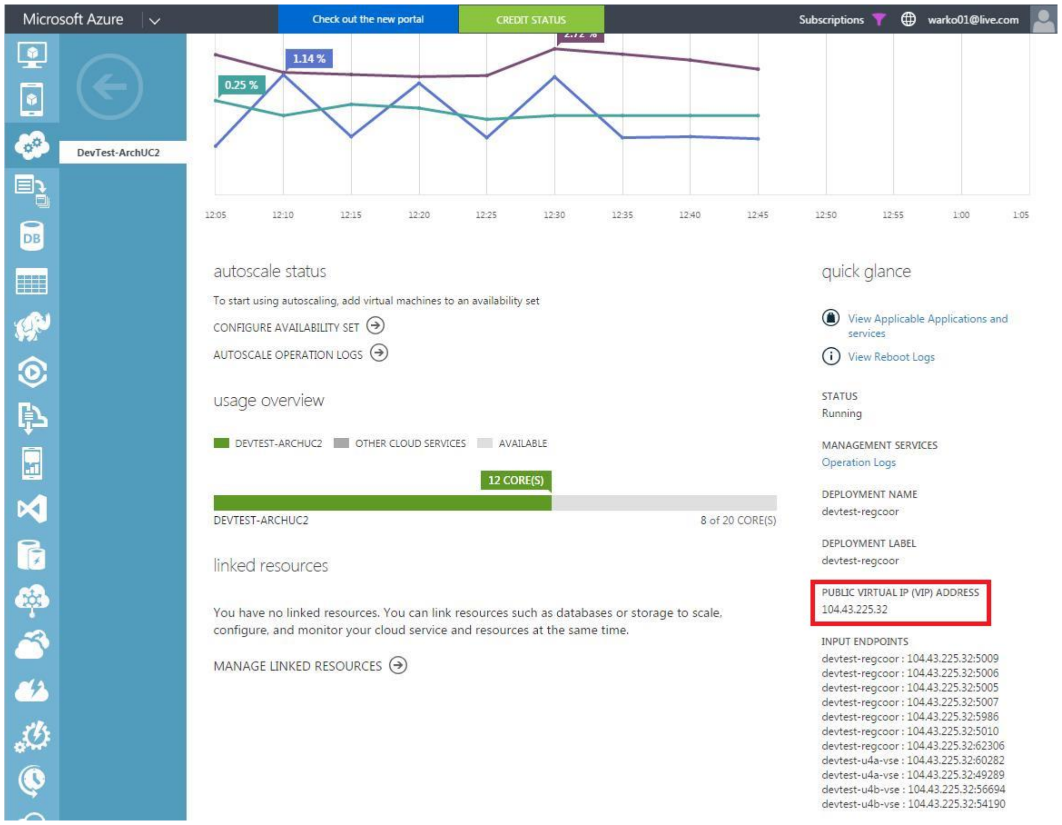
Figure 2 - Cloud Service - VMs and Resources

The cloud service in Microsoft® Azure for DevTest acts like a router to the subnet where the DevTest servers and components are running.

Network

As mentioned before, public remote access to the DevTest environment in Microsoft® Azure is available through the public IP address of Azure's cloud service specific for the DevTest deployment in Azure. Microsoft® Azure assigns a public IP address to the cloud service dynamically upon its start. (Classic)

Azure Portal publishes the cloud service's public IP address. Please be aware that the IP address may change between restarts of the cloud service



autoscale status

To start using autoscaling, add virtual machines to an availability set

CONFIGURE AVAILABILITY SET

AUTOSCALE OPERATION LOGS

usage overview

🟩 DEVTEST-ARCHUC2 ▨ OTHER CLOUD SERVICES ⬜ AVAILABLE

12 CORE(S)

DEVTEST-ARCHUC2 8 of 20 CORE(S)

linked resources

You have no linked resources. You can link resources such as databases or storage to scale, configure, and monitor your cloud service and resources at the same time.

MANAGE LINKED RESOURCES

quick glance

View Applicable Applications and services

View Reboot Logs

STATUS
Running

MANAGEMENT SERVICES
Operation Logs

DEPLOYMENT NAME
devtest-regcoor

DEPLOYMENT LABEL
devtest-regcoor

PUBLIC VIRTUAL IP (VIP) ADDRESS
104.43.225.32

INPUT ENDPOINTS
devtest-regcoor : 104.43.225.32:5009
devtest-regcoor : 104.43.225.32:5006
devtest-regcoor : 104.43.225.32:5005
devtest-regcoor : 104.43.225.32:5007
devtest-regcoor : 104.43.225.32:5986
devtest-regcoor : 104.43.225.32:5010
devtest-regcoor : 104.43.225.32:62306
devtest-u4a-vse : 104.43.225.32:60282
devtest-u4a-vse : 104.43.225.32:49289
devtest-u4b-vse : 104.43.225.32:56694
devtest-u4b-vse : 104.43.225.32:54190

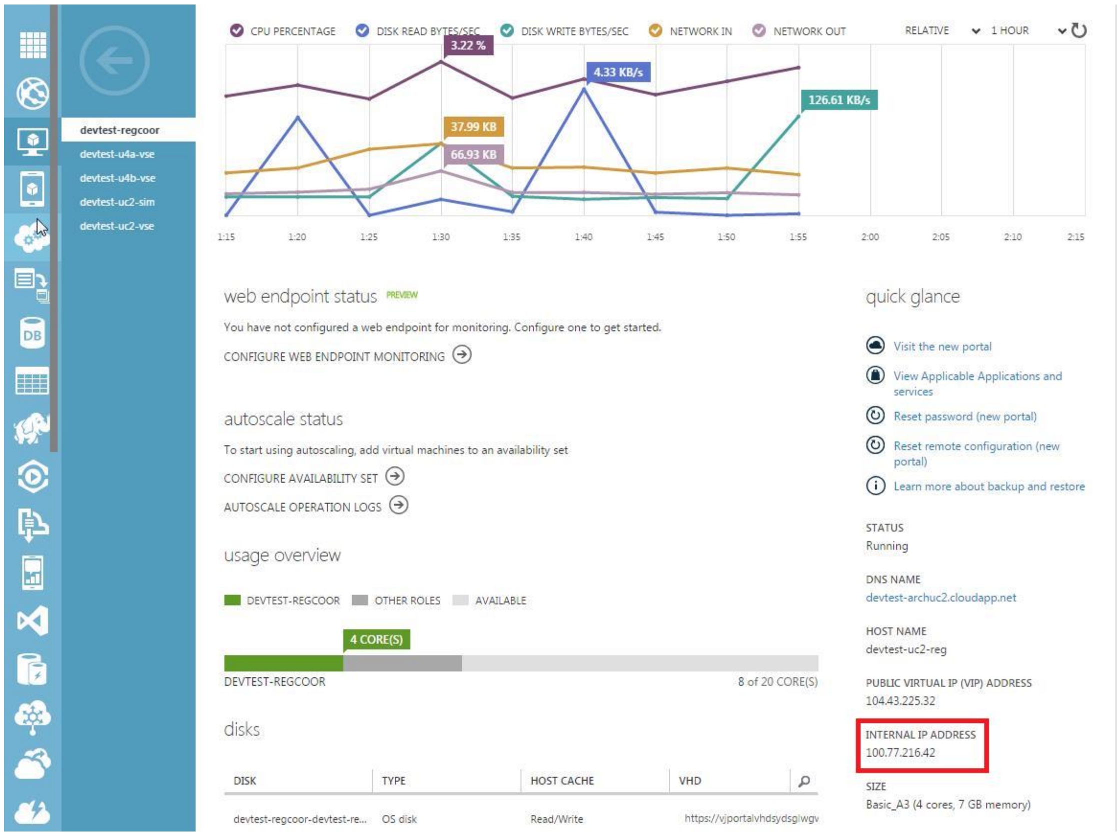
Figure 3 - Cloud Service – Public IP address

Virtual Machines

The DevTest Server components run in Microsoft® Azure vitual machines. These are configured with following resources:

Host name	Private IP	Size	#NICs
devtest-uc2-reg	<dynamic>	Basic_A3	1 NIC
devtest-uc2-sim	<dynamic>	Basic_A2	1 NIC
devtest-u4a-sim	<dynamic>	Basic_A2	1 NIC
devtest-u4b-sim	<dynamic>	Basic_A2	1 NIC
devtest-uc2-vse	<dynamic>	Basic_A2	1 NIC
devtest-u4a-vse	<dynamic>	Basic_A2	1 NIC
devtest-u4b-vse	<dynamic>	Basic_A2	1 NIC

Microsoft® Azure assigns private IP addresses dynamically upon VM startup. Microsoft® Azure Portal displays a VM's private IP address:



web endpoint status PREVIEW

You have not configured a web endpoint for monitoring. Configure one to get started.

CONFIGURE WEB ENDPOINT MONITORING

autoscale status

To start using autoscaling, add virtual machines to an availability set

CONFIGURE AVAILABILITY SET

AUTOSCALE OPERATION LOGS

usage overview

🟩 DEVTEST-REGCOOR ▨ OTHER ROLES ▨ AVAILABLE

4 CORE(S)

DEVTEST-REGCOOR 8 of 20 CORE(S)

disks

DISK	TYPE	HOST CACHE	VHD
devtest-regcoor-devtest-re...	OS disk	Read/Write	https://vjportalvhdsydsglwgv

quick glance

Visit the new portal

View Applicable Applications and services

Reset password (new portal)

Reset remote configuration (new portal)

Learn more about backup and restore

STATUS
Running

DNS NAME
devtest-archuc2.cloudapp.net

HOST NAME
devtest-uc2-reg

PUBLIC VIRTUAL IP (VIP) ADDRESS
104.43.225.32

INTERNAL IP ADDRESS
100.77.216.42

SIZE
Basic_A3 (4 cores, 7 GB memory)

Figure 4 - VM private IP address

In current Microsoft® Azure version, the VM sizes represent following resources:

Size Type	Cores	RAM	#Disk	Disk size
Basic_A3	4	7 GB	2	130 GB / 120 GB
Basic_A2	2	3.5 GB	2	130 GB / 60 GB

Network Ports

The cloud service network blocks any inbound network traffic that is not required to access the provided systems. Therefore, any ports needed to access DevTest components must be published in Microsoft® 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.

Remote Desktop Protocol

Remote Desktop Access (RDP) to the DevTest systems in Microsoft® Azure is available through the public IP address of the DevTest cloud service and the VM specific RDP port number. Microsoft® Azure creates the VM specific ports as part of the VM provisioning process.

Service	Host name	Public IP
Enterprise Dashboard, Registry, Broker, Coordinator, Portal	devtest-uc2-reg	<Cloud service>:62306
Sim-UC2	devtest-uc2-sim	<Cloud service>:54750
Sim-U4A	devtest-u4a-sim	<Cloud service>:56590
Sim-U4B	devtest-u4b-sim	<Cloud service>:57779
VSE-UC2	devtest-uc2-vse	<Cloud service>:57151
VSE-UC4A	devtest-u4a-vse	<Cloud service>:55573
VSE-UC4B	devtest-u4b-vse	<Cloud service>:54190
Visual SVN	devtest-uc2-sim	<Cloud service>:54750

Powershell

By default, Microsoft® Azure also creates a individual and separate port number for Powershell access to each VM.

For other services that users need remote access to from Internet, such as some of the DevTest services, the DevTest administrator of the Azure environment has to create additional port mappings.

HTTP/TCP Ports exposed for each of the DevTest Components

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

Inbound Access

Following virtual machines require external access to services, which need to be configured in Microsoft® Azure:

Server	VM name
Registry/Coordinator/Portal	devtest-uc2-reg
Simulator-UC2	devtest-uc2-sim
Simulator-U4A	devtest-u4a-sim
Simulator-U4B	devtest-u4b-sim
VSE-UC2	devtest-uc2-vse
VSE-U4A	devtest-u4a-vse

VSE-U4B	devtest-u4b-vse

Following services on these virtual machines running Microsoft® Windows Server need inbound access for remote users:

- Windows Services on all virtual machines

Service name	Default Port no.
Remote Desktop	3389
PowerShell	5986

- DevTest Services
  - Registry Server

Service name	Default Port no.
Registry	2010
Coordinator	2011
Server Console Web UI	1505
Enterprise Dashboard UI	1506
Portal UI	1507
LISA Bank	8080

  - VSE Servers need inbound access configured for any deployed virtual service

Outbound Access

When installing DevTest components on VMs in the cloud, DevTest Registry services access the DevTest Enterprise Dashboard service. Therefore, outbound access to the Enterprise Dashboard service from Microsoft® Azure VMs to Enterprise Dashboard service on customer premise on default port 2003 must be available.

Following services on these virtual machines running Microsoft® Windows Server may need outbound access:

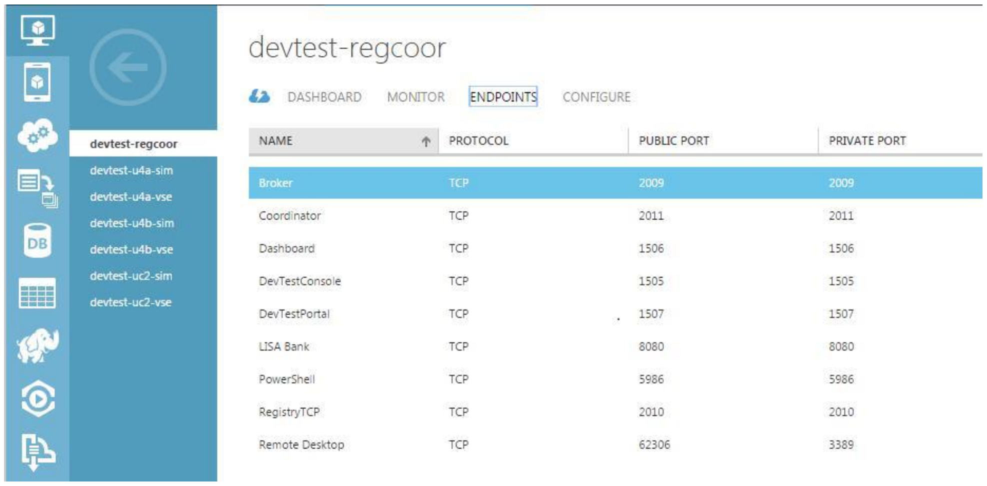
- DevTest Services

Service name	Default Port no.
Enterprise Dashboard	2003

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

Registry Server

The common DevTest services are installed on – what we call – the Registry server *devtest-uc2-reg*. This server hosts the Enterprise Dashboard, the Registry, the Coordinator and the broker service. Most of those services must be accessible remotely by other services or User Interfaces. Therefore, public ports are assigned to those services on this system that need to be open for remote access.



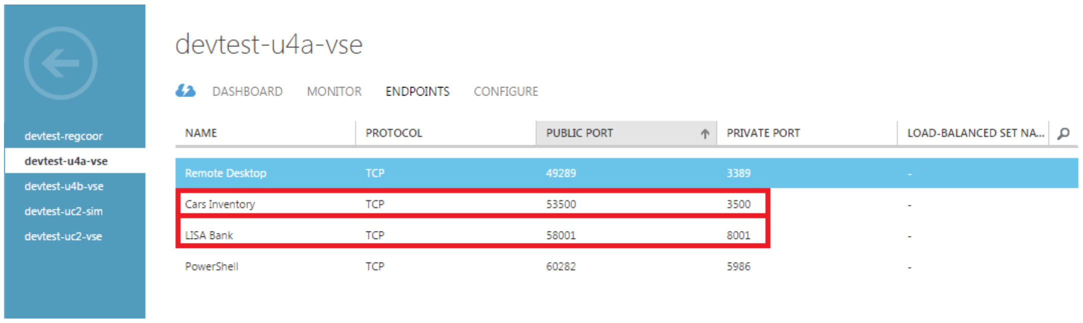
NAME	PROTOCOL	PUBLIC PORT	PRIVATE PORT
Broker	TCP	2009	2009
Coordinator	TCP	2011	2011
Dashboard	TCP	1506	1506
DevTestConsole	TCP	1505	1505
DevTestPortal	TCP	1507	1507
LISA Bank	TCP	8080	8080
PowerShell	TCP	5986	5986
RegistryTCP	TCP	2010	2010
Remote Desktop	TCP	62306	3389

Figure 5 - Registry Server Public Port Mapping

VSE Server

As an example, two public ports are configured to open public access to popular virtual demo services on this sample VSE. In case of recording a LISA Bank session on port 8001 and a subsequent deployment of the created virtual service to this VSE server, public port 58001 is open to access this virtual service.

Similarly, if a virtual service is created for the Cars Inventory service of the DevTest Cars demo based on Continuous Application Insight (CAI), and deployed to this VSE server, public port 53500 is open to access this virtual service.



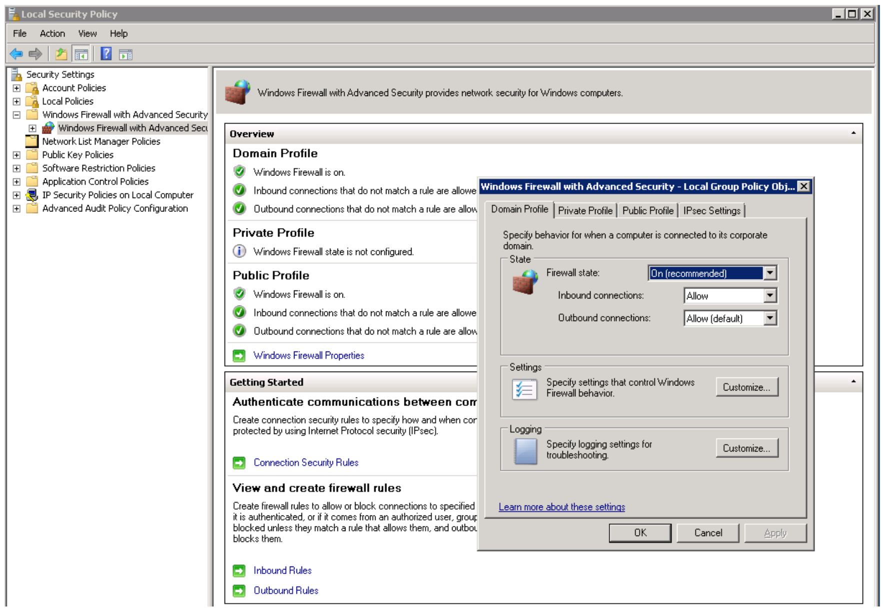
NAME	PROTOCOL	PUBLIC PORT	PRIVATE PORT	LOAD-BALANCED SET NA...
Remote Desktop	TCP	49289	3389	-
Cars Inventory	TCP	53500	3500	-
LISA Bank	TCP	58001	8001	-
PowerShell	TCP	60282	5986	-

Figure 6 - VSE Server Public Port Mapping

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


Local Security Policy

File Action View Help

Security Settings
- Account Policies
- Local Policies
- Windows Firewall with Advanced Security
  - Windows Firewall with Advanced Secu
- Network List Manager Policies
- Public Key Policies
- Software Restriction Policies
- Application Control Policies
- IP Security Policies on Local Computer
- Advanced Audit Policy Configuration

Windows Firewall with Advanced Security provides network security for Windows computers.

Overview

Domain Profile
- Windows Firewall is on.
- Inbound connections that do not match a rule are allowe
- Outbound connections that do not match a rule are allow

Private Profile
- Windows Firewall state is not configured.

Public Profile
- Windows Firewall is on.
- Inbound connections that do not match a rule are allowe
- Outbound connections that do not match a rule are allow

Windows Firewall Properties

Getting Started

Authenticate communications between com

Create connection security rules to specify how and when cor protected by using Internet Protocol security (IPsec).

Connection Security Rules

View and create firewall rules

Create firewall rules to allow or block connections to specified it is authenticated, or if it comes from an authorized user, grour blocked unless they match a rule that allows them, and outbou blocks them.

Inbound Rules

Outbound Rules

Windows Firewall with Advanced Security - Local Group Policy Obj...

Domain Profile | Private Profile | Public Profile | IPsec Settings

Specify behavior for when a computer is connected to its corporate domain.

State

Firewall state:	On (recommended)
Inbound connections:	Allow
Outbound connections:	Allow (default)

Settings

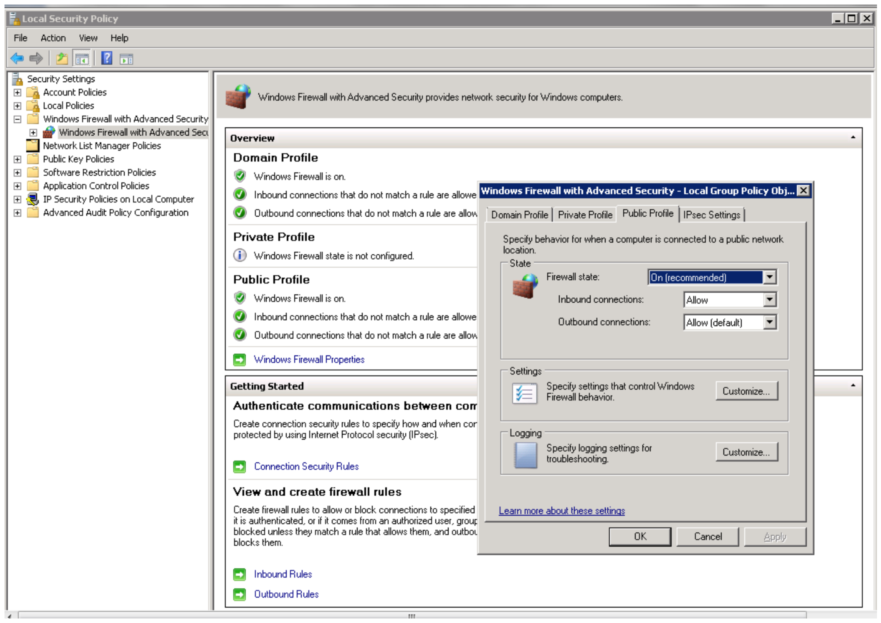
Specify settings that control Windows Firewall behavior. Customize...

Logging

Specify logging settings for troubleshooting. Customize...

Learn more about these settings

OK Cancel Apply


Local Security Policy

File Action View Help

Security Settings
- Account Policies
- Local Policies
- Windows Firewall with Advanced Security
  - Windows Firewall with Advanced Secu
- Network List Manager Policies
- Public Key Policies
- Software Restriction Policies
- Application Control Policies
- IP Security Policies on Local Computer
- Advanced Audit Policy Configuration

Windows Firewall with Advanced Security provides network security for Windows computers.

Overview

Domain Profile
- Windows Firewall is on.
- Inbound connections that do not match a rule are allowe
- Outbound connections that do not match a rule are allow

Private Profile
- Windows Firewall state is not configured.

Public Profile
- Windows Firewall is on.
- Inbound connections that do not match a rule are allowe
- Outbound connections that do not match a rule are allow

Windows Firewall Properties

Getting Started

Authenticate communications between com

Create connection security rules to specify how and when cor protected by using Internet Protocol security (IPsec).

Connection Security Rules

View and create firewall rules

Create firewall rules to allow or block connections to specified it is authenticated, or if it comes from an authorized user, grour blocked unless they match a rule that allows them, and outbou blocks them.

Inbound Rules

Outbound Rules

Windows Firewall with Advanced Security - Local Group Policy Obj...

Domain Profile | Private Profile | Public Profile | IPsec Settings

Specify behavior for when a computer is connected to a public network location.

State

Firewall state:	On (recommended)
Inbound connections:	Allow
Outbound connections:	Allow (default)

Settings

Specify settings that control Windows Firewall behavior. Customize...

Logging

Specify logging settings for troubleshooting. Customize...

Learn more about these settings

OK Cancel Apply

DevTest configuration

DevTest 9.0 is installed with default settings. Particularly, when installing the distributed service components VSE and Simulator as Windows services on dedicated VMs, DevTest server components were configured not to start automatically on system start.

In a post-install step Windows services VSE and Simulator on the dedicated VMs were reconfigured to start automatically on system startup after local.properties file was modified to adjust the DevTest configuration accordingly.

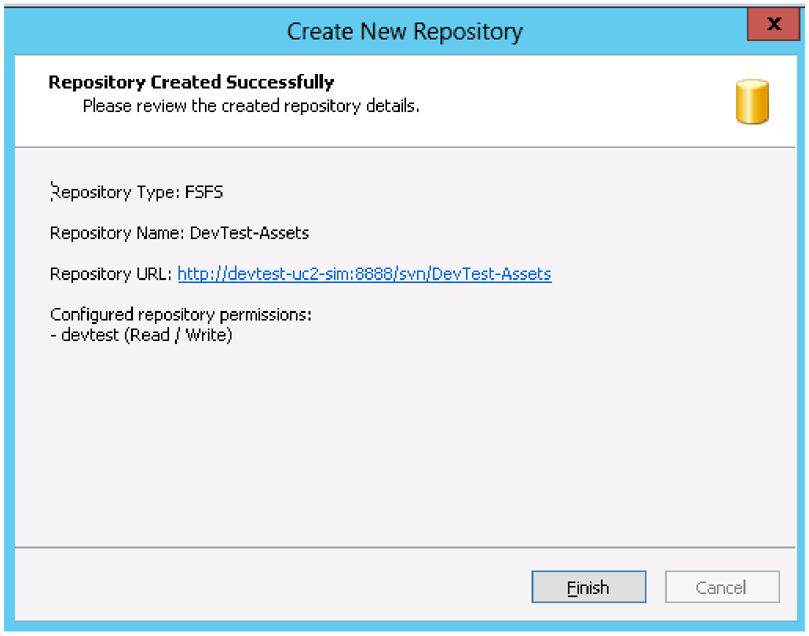
On the VMs dedicated for running VSE and Simulator services, DevTest's local.properties files need adjustments:

1. As there is no local DevTest Registry service, VSE service needs information where to find the Registry service to connect. We enhance file **local.properties** by
   `lisa.registry.url=tcp://devtest-uc2-reg:2010/Registry`
   Notes:
   - We do not use the IP address to access the Registry service, but the VM's hostname. This saves us updating local.properties each time after restart. Microsoft Azure takes care of name resolutiuon.
2. Because DevTest portal displays VSEs and Simulators by name, these names need to be distinct. By default, all VSE services are called 'VSE', and all Simulator services go by 'Simulator'. We use an identifying substring of the system's hostname to identify the VSE or Simulator service, respectively, in the various UIs.
   a. On server *devtest-u4a-vse* we add the distinct VSE name to **local.properties**
      `lisa.vseName=VSE-U4A`
   b. On server *devtest-uc2-sim* we add the distinct Simulator name to **local.properties**
      `lisa.vseName=Sim-UC2`

SCM Details

This section shows how SCM is used –

- A new repository was created in Visual SVN Server called DevTest-Assets



Repository Created Successfully
Please review the created repository details.

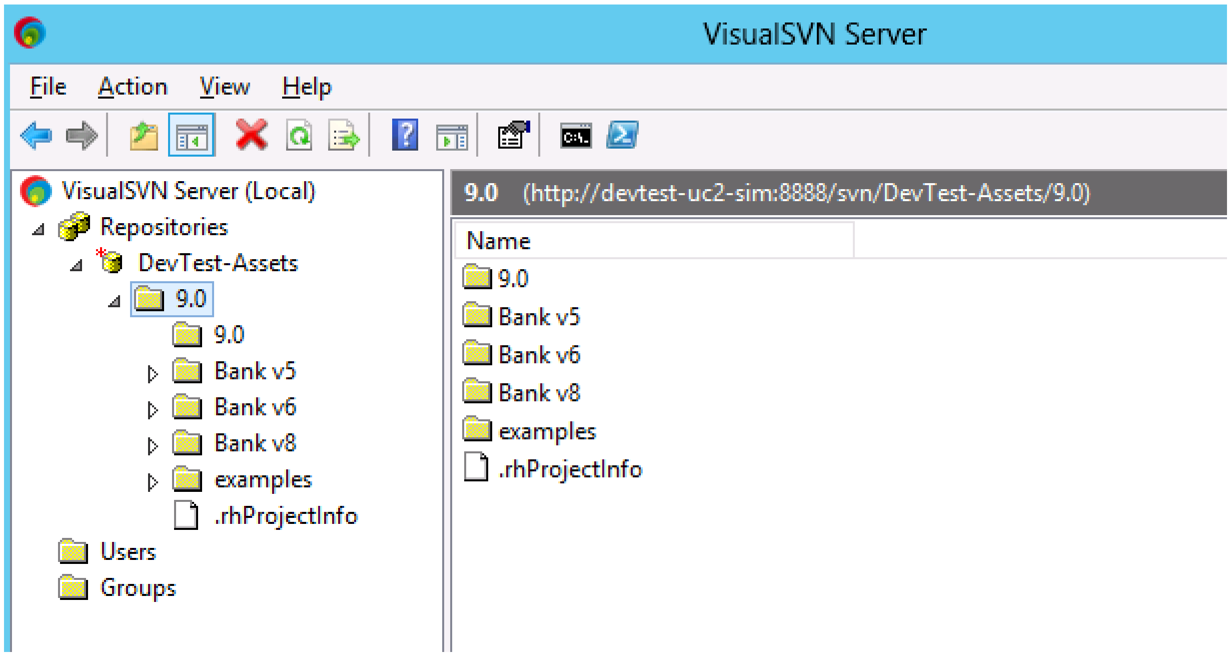
Repository Type: FSFS

Repository Name: DevTest-Assets

Repository URL: http://devtest-uc2-sim:8888/svn/DevTest-Assets

Configured repository permissions:
- devtest (Read / Write)

- From the location of the DevTest "Projects" folder, the Tortoise SVN client was used to import all the Project assets into SVN's DevTest-Assets repository



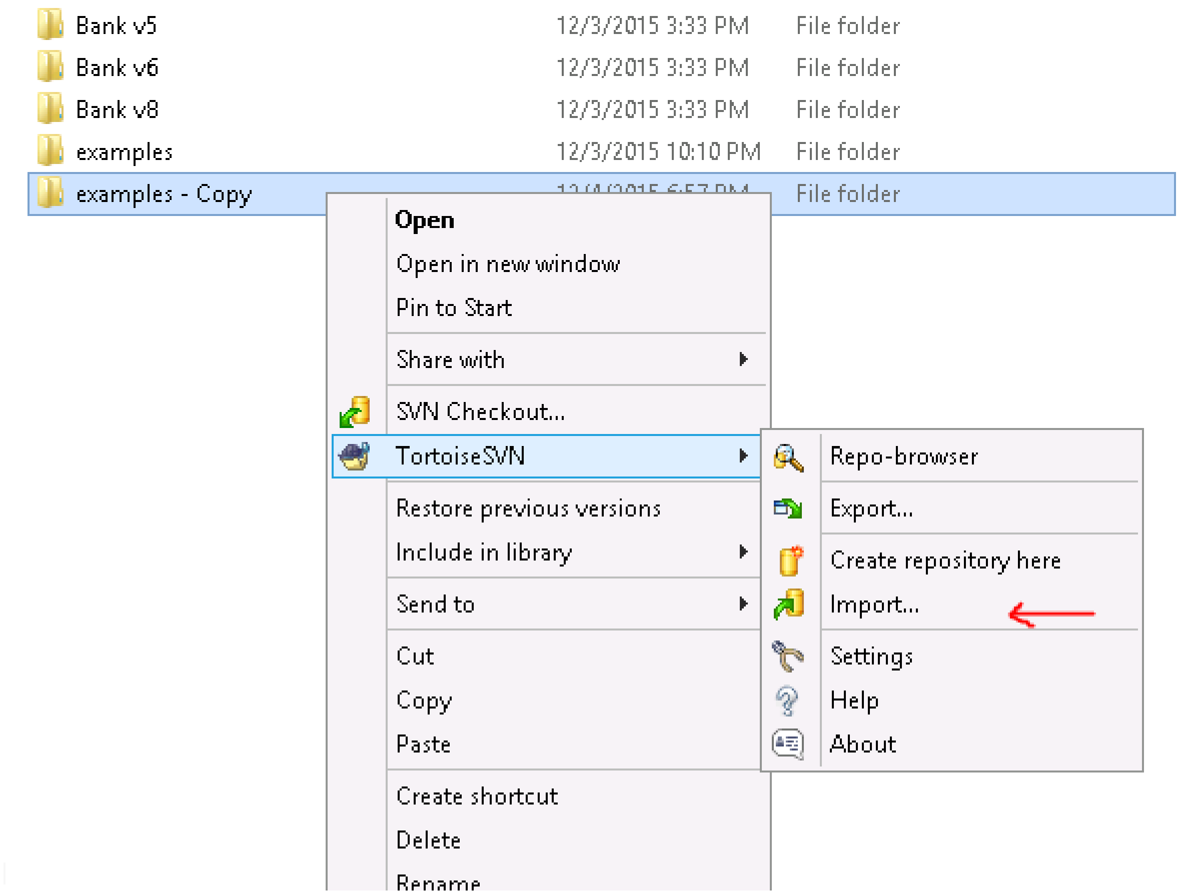
VisualSVN Server (Local)
- Repositories
  - DevTest-Assets
    - 9.0
      - 9.0
      - Bank v5
      - Bank v6
      - Bank v8
      - examples
      - .rhProjectInfo
- Users
- Groups

9.0 (http://devtest-uc2-sim:8888/svn/DevTest-Assets/9.0)

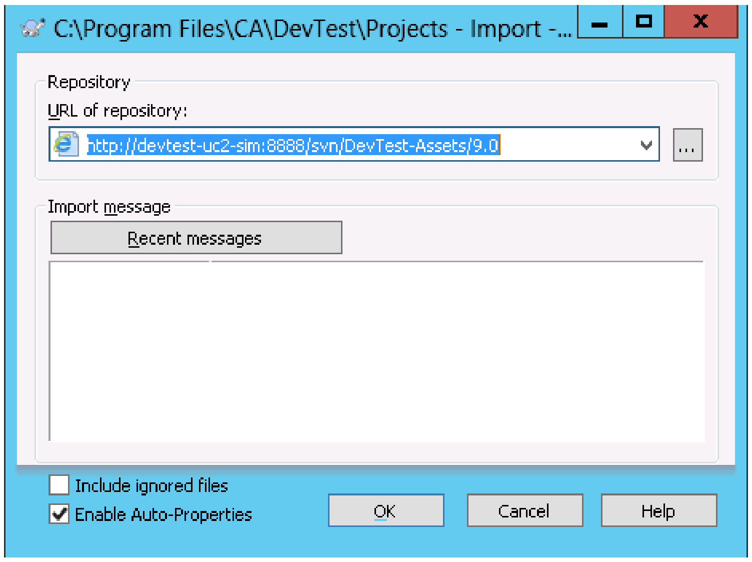
Name
9.0
Bank v5
Bank v6
Bank v8
examples
.rhProjectInfo

- At this point, the assumption is that the DevTest users have a mapped drive to the DevTest Projects folder. Once they create/update a new DevTest asset, they can simply commit those changes to the SVN using the Tortoise client.
- In this example, this end user would like to back up the ‘examples’ folder as ‘examples - copy’. After copy/paste, a new folder called ‘examples – copy’ is created.

- Then the user right clicks and proceeds to check in the new folder to SVN as shown below



Authenticate as required –

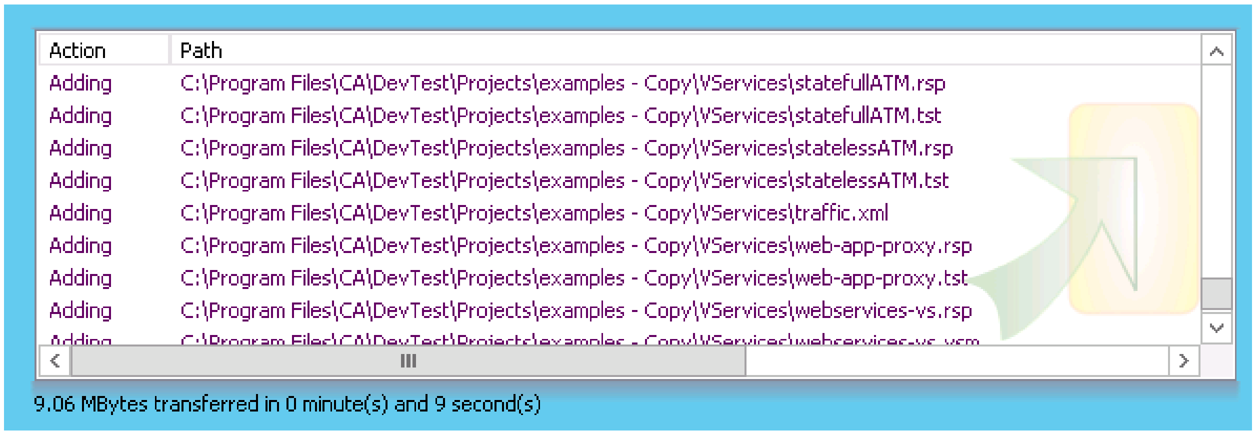


URL of repository: http://devtest-uc2-sim:8888/svn/DevTest-Assets/9.0

☐ Include ignored files

☑ Enable Auto-Properties

Completed check in to SVN server shown



Action	Path
Adding	C:\Program Files\CA\DevTest\Projects\examples - Copy\VServices\statefullATM.rsp
Adding	C:\Program Files\CA\DevTest\Projects\examples - Copy\VServices\statefullATM.tst
Adding	C:\Program Files\CA\DevTest\Projects\examples - Copy\VServices\statelessATM.rsp
Adding	C:\Program Files\CA\DevTest\Projects\examples - Copy\VServices\statelessATM.tst
Adding	C:\Program Files\CA\DevTest\Projects\examples - Copy\VServices\traffic.xml
Adding	C:\Program Files\CA\DevTest\Projects\examples - Copy\VServices\web-app-proxy.rsp
Adding	C:\Program Files\CA\DevTest\Projects\examples - Copy\VServices\web-app-proxy.tst
Adding	C:\Program Files\CA\DevTest\Projects\examples - Copy\VServices\webservices-vs.rsp

```
9.06 MBytes transferred in 0 minute(s) and 9 second(s)
```

A new folder is created in the SVN server for ‘examples – copy’.

In this way, the same technique can be applied to any DevTest assets that are created/modified.

If you were using the DevTest Workstation and the assets were stored locally then you would check in and check out assets from the SVN server without the need to map to the centrally located DevTest Projects folder.

Setup Verification

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

- Determine that all components are connected to the DevTest Registry service
- Verify that test cases can be launched from the DevTest Portal
- Verify that virtual services can be deployed from the DevTest Portal
- Check if tests can be monitored in DevTest Portal
- Check if virtual services show up in DevTest Portal’s VSE
- Check if local Workstation connects to the Registry service in Microsoft® Azure
- Check if local test client (LISA Bank kiosk) connects to application in Microsoft® Azure
- Check if local test client (LISA Bank kiosk) connects to virtual service in Microsoft® Azure
- Check if local test case can be staged to coordinator in Microsoft® Azure

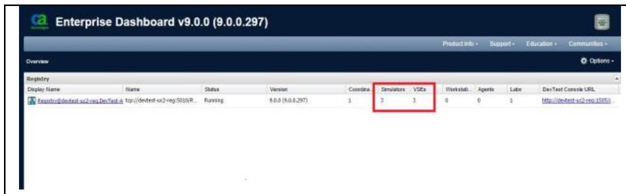
These verification steps are executed from the Enterprise Dashboard and the DevTest Portal to test the various access methods.

DevTest Portal (from within the company network)

DevTest Portal is the Web UI frontend 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

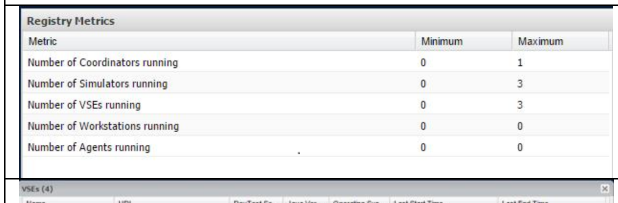
Component Connectivity Check



Display Name	Name	Status	Version	Coordina...	Simulators	VSEs	Workstati...	Agents	Labs	DevTest Console URL
Registry@devtest-uc2-reg.DevTest-A	tcp://devtest-uc2-reg:5010/R...	Running	9.0.0 (9.0.0.297)	1	3	3	0	0	1	http://devtest-uc2-reg:1505/i...

Enterprise Dashboard shows all the components that have connected to the registry with the following components running in the Azure environment

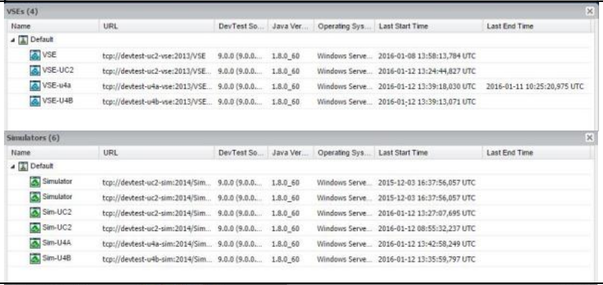
- Registry
- Coordinator
- 3 Simulators
- 3 VSEs
- Workstations



Registry Metrics

Metric	Minimum	Maximum
Number of Coordinators running	0	1
Number of Simulators running	0	3
Number of VSEs running	0	3
Number of Workstations running	0	0
Number of Agents running	0	0

The detailed view in Enterprise Dashboard shows the list of the various services, i.e. the several VSE and Simulator services.



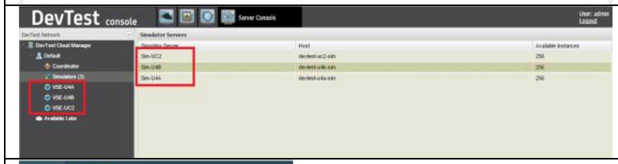
VSEs (4)

Name	URL	DevTest So...	Java Ver...	Operating Sys...	Last Start Time	Last End Time
Default						
VSE	tcp://devtest-uc2-vse:2013/VSE	9.0.0 (9.0.0....	1.8.0_60	Windows Serve...	2016-01-08 13:58:13,784 UTC	
VSE-UC2	tcp://devtest-uc2-vse:2013/VSE...	9.0.0 (9.0.0....	1.8.0_60	Windows Serve...	2016-01-12 13:24:44,827 UTC	
VSE-u4a	tcp://devtest-u4a-vse:2013/VSE...	9.0.0 (9.0.0....	1.8.0_60	Windows Serve...	2016-01-12 13:39:18,030 UTC	2016-01-11 10:25:20,975 UTC
VSE-U4B	tcp://devtest-u4b-vse:2013/VSE...	9.0.0 (9.0.0....	1.8.0_60	Windows Serve...	2016-01-12 13:39:13,071 UTC	

Simulators (6)

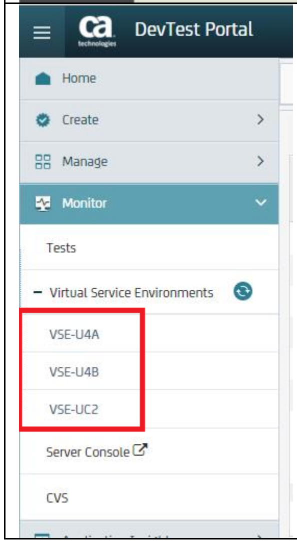
Name	URL	DevTest So...	Java Ver...	Operating Sys...	Last Start Time	Last End Time
Default						
Simulator	tcp://devtest-uc2-sim:2014/Sim...	9.0.0 (9.0.0....	1.8.0_60	Windows Serve...	2015-12-03 16:37:56,057 UTC	
Simulator	tcp://devtest-uc2-sim:2014/Sim...	9.0.0 (9.0.0....	1.8.0_60	Windows Serve...	2015-12-03 16:37:56,057 UTC	
Sim-UC2	tcp://devtest-uc2-sim:2014/Sim...	9.0.0 (9.0.0....	1.8.0_60	Windows Serve...	2016-01-12 13:27:07,695 UTC	
Sim-UC2	tcp://devtest-uc2-sim:2014/Sim...	9.0.0 (9.0.0....	1.8.0_60	Windows Serve...	2016-01-12 08:55:32,237 UTC	
Sim-U4A	tcp://devtest-u4a-sim:2014/Sim...	9.0.0 (9.0.0....	1.8.0_60	Windows Serve...	2016-01-12 13:42:58,249 UTC	
Sim-U4B	tcp://devtest-u4b-sim:2014/Sim...	9.0.0 (9.0.0....	1.8.0_60	Windows Serve...	2016-01-12 13:35:59,797 UTC	

The report view on VSEs and Simulators displays the log of start and stop times of VSEs and Simulator services, respectively.



Simulator Server	Host	Available Instances
Sim-UC2	devtest-uc2-sim	256
Sim-U4B	devtest-u4b-sim	256
Sim-U4A	devtest-u4a-sim	256

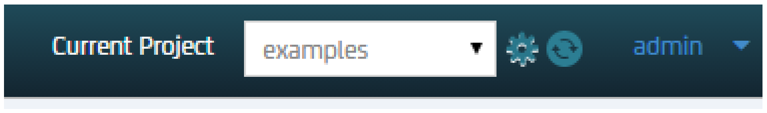
The Server Console also displays all 3 VSEs and Simulators



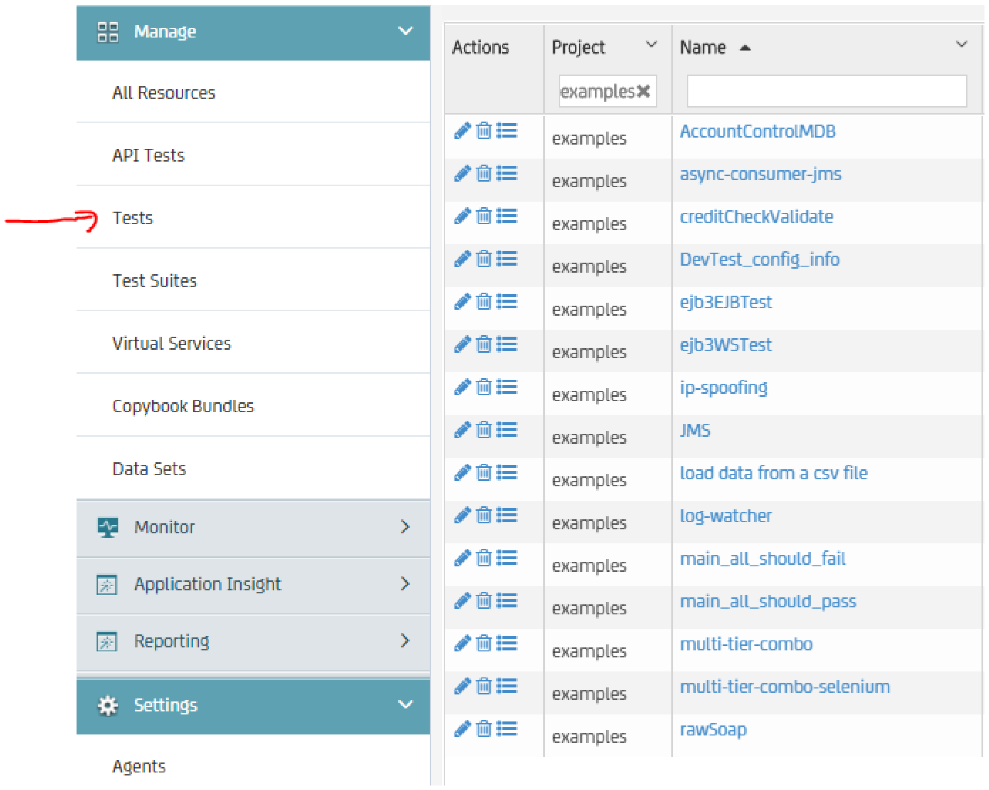
The Portal also displays all the three VSEs

Running Tests from DevTest Portal

In the DevTest Portal, select the 'examples' folder. *Note – to view the 'examples' folder in the DevTest Portal, you would need to copy that folder to the Projects folder*.

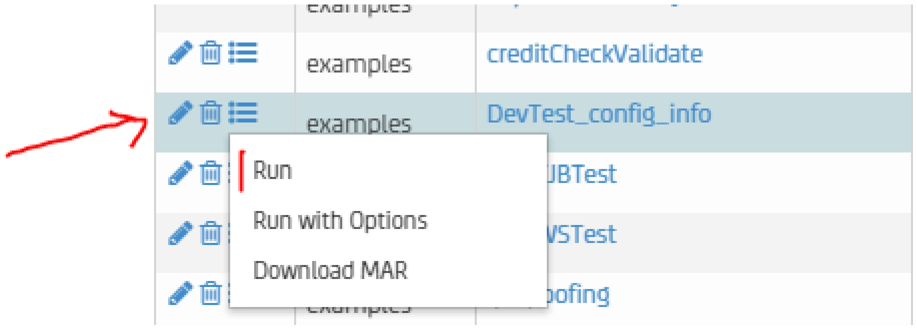


In the left menu – select Manage → Tests

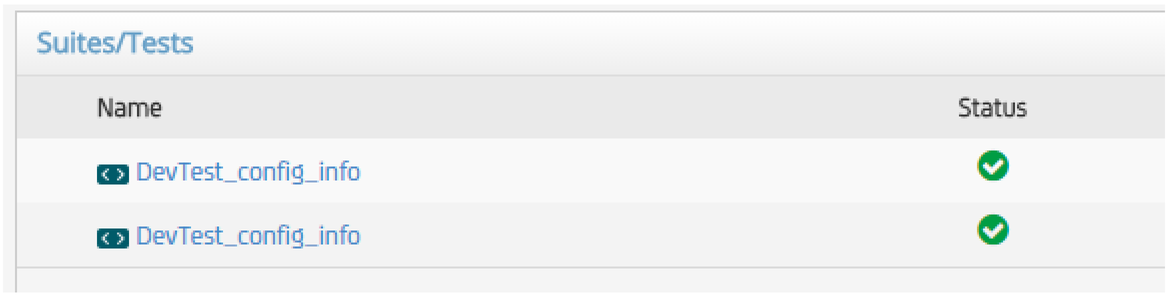


Actions	Project	Name
	examples	
	examples	AccountControlMDB
	examples	async-consumer-jms
	examples	creditCheckValidate
	examples	DevTest_config_info
	examples	ejb3EJBTest
	examples	ejb3WSTest
	examples	ip-spoofing
	examples	JMS
	examples	load data from a csv file
	examples	log-watcher
	examples	main_all_should_fail
	examples	main_all_should_pass
	examples	multi-tier-combo
	examples	multi-tier-combo-selenium
	examples	rawSoap

You will now locate and run the DevTest config_info test case.



As soon as you click Run, a new tab called ‘Monitoring Tests’ is opened. The test should run to completion.



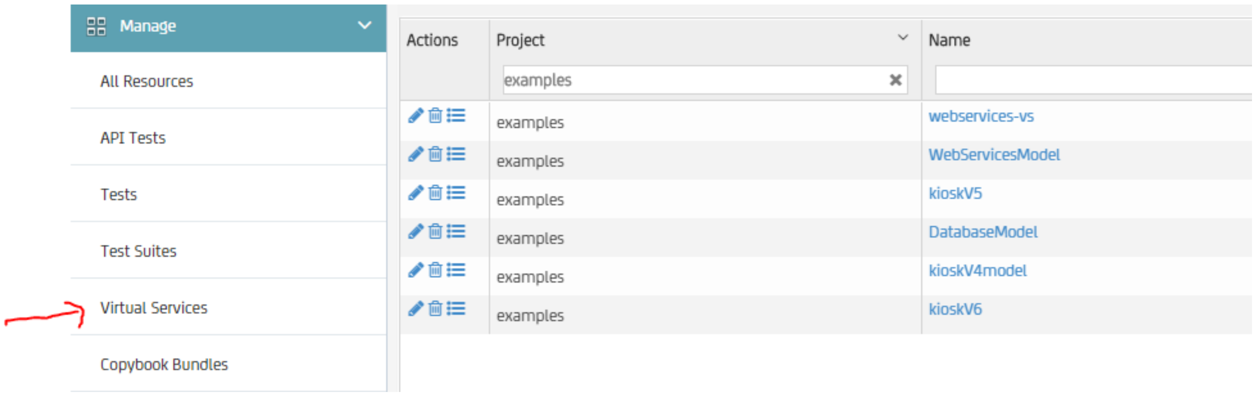
Suites/Tests

Name	Status
DevTest_config_info	✓
DevTest_config_info	✓

You can now click on the completed test case name to drill down into the details of the execution steps.

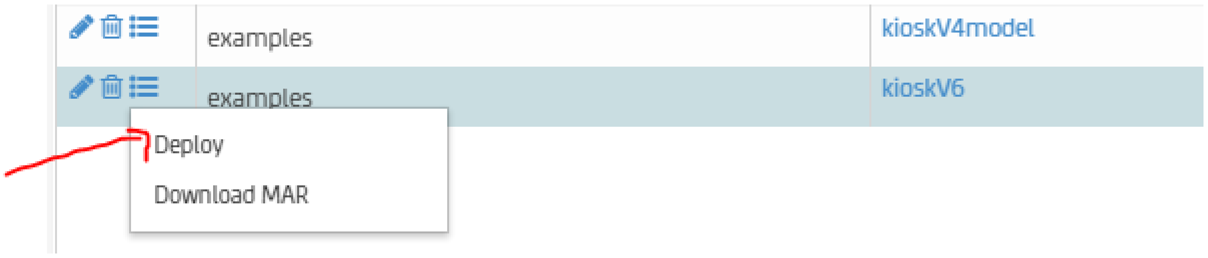
Deploy Virtual Service to VSE from the DevTest Portal

In the left menu, click on Manage → Virtual Services to open a list of available virtual services in the examples folder.

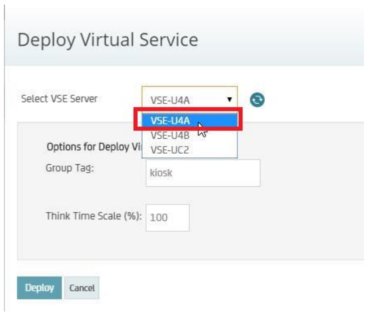


Actions	Project	Name
	examples	
	examples	webservices-vs
	examples	WebServicesModel
	examples	kioskV5
	examples	DatabaseModel
	examples	kioskV4model
	examples	kioskV6

Deploy the virtual service called as kioskV6 by clicking on the Options action and selecting Deploy



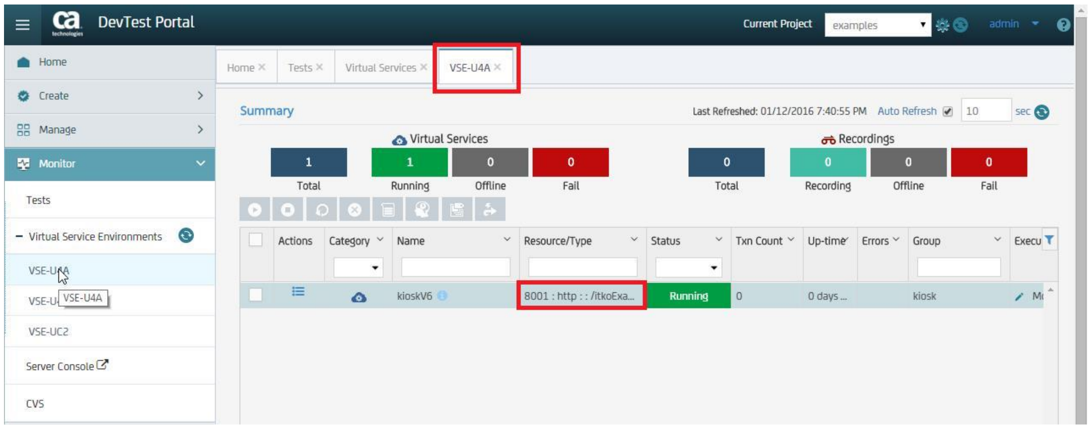
Pick any VSE server and provide an optional Group tag. You should see a message about successful deployment. We pick VSE-U4A, because we added a public port to the cloud service to access this VSE on port 8001.



Deploy Virtual Service

Select VSE Server: VSE-U4A

Now click on Monitor →Virtual Service Environment → VSE-U4A. This will open a new tab called VSE-U4A and you should see the kioskV6 VS deployed and running successfully on port 8001.



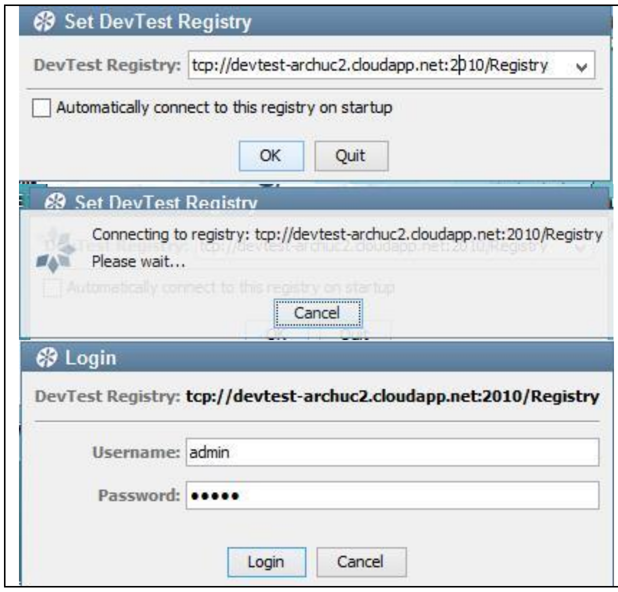
Last Refreshed: 01/12/2016 7:40:55 PM Auto Refresh ✓ 10 sec

Virtual Services: Total	Running	Offline	Fail	Recordings: Total	Recording	Offline	Fail
1	1	0	0	0	0	0	0

Actions	Category	Name	Resource/Type	Status	Txn Count	Up-time	Errors	Group	Execu
		kioskV6	8001 : http : : /itkoExa...	Running	0	0 days ...		kiosk	M

DevTest Workstation (from within the company network)

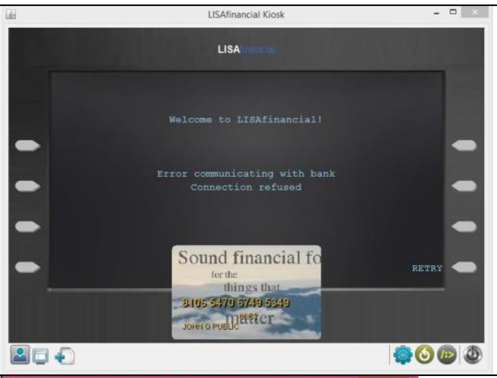
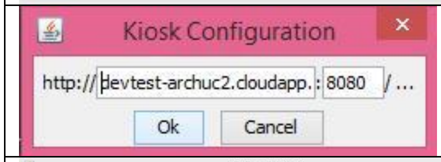
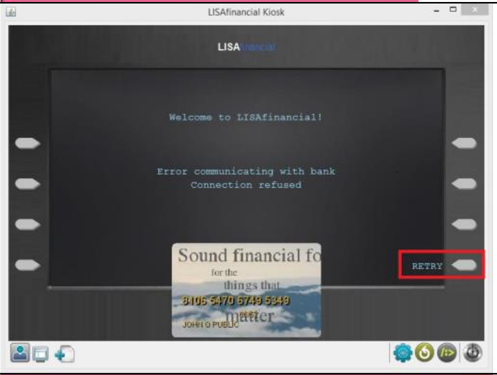
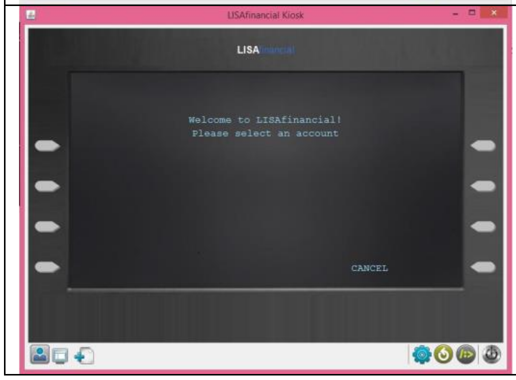
First we verify that the local workstation on my company network is able to access the Microsoft® Azure environment.

 Set DevTest Registry — DevTest Registry: tcp://devtest-archuc2.cloudapp.net:2010/Registry; ☐ Automatically connect to this registry on startup; OK Quit. Set DevTest Registry — Connecting to registry: tcp://devtest-archuc2.cloudapp.net:2010/Registry Please wait... Cancel. Login — DevTest Registry: tcp://devtest-archuc2.cloudapp.net:2010/Registry; Username: admin; Password: •••••; Login Cancel	Connect local workstation on company network to registry on Microsoft® Azure cloud service *devtest-archu2.cloudapp.net:2010*.

Connect local client to Application in cloud

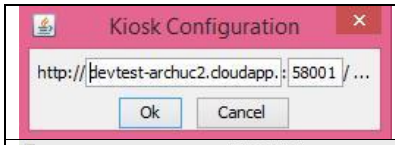
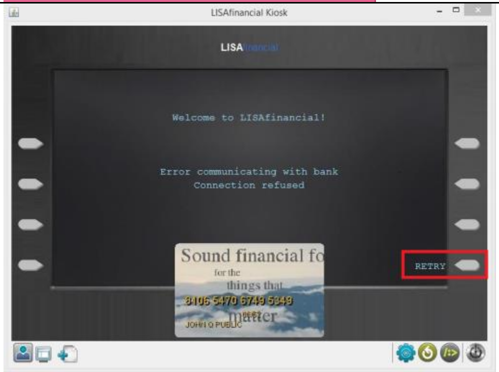
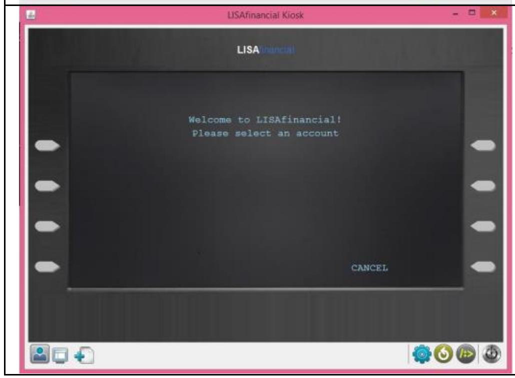
Now we want to verify that my local test client can connect to the application in the Microsoft® Azure environment.

Start LISA Bank demo on *devtest-uc2-reg*. Public port *devtest-archu2.cloudapp.net:8080*allows access to *devtest-uc2-reg:8080*, which is the port of the LISA Bank service.

 LISAfinancial Kiosk LISAfinancial RETRY Sound financial fo for the things that matter 8106 5470 6749 5349 JOHN Q PUBLIC Welcome to LISAfinancial! Error communicating with bank Connection refused	Start Kiosk from local Workstation (on Company network). Note the failed connection to the bank.
 http:// devtest-archuc2.cloudapp. : 8080 / …	Edit the URL to access LISA bank to *devtest-archuc2.cloudapp.net:8080*
 LISAfinancial Kiosk LISAfinancial RETRY Sound financial fo for the things that matter 8106 5470 6749 5349 JOHN Q PUBLIC Welcome to LISAfinancial! Error communicating with bank Connection refused	Click on the Retry button to apply the changed URL …
 Welcome to LISAfinancial! Please select an account	… and the kiosk on local workstation is now connected to the LISA Bank application on *devtest-uc2-reg* in Microsoft® Azure. Click on the credit card to select a user, try Lisa Simpson with password golisa, which will log you in. The screenshot on the left shows a successful log in of Lisa Simpson.

Connect local client to virtual service in cloud

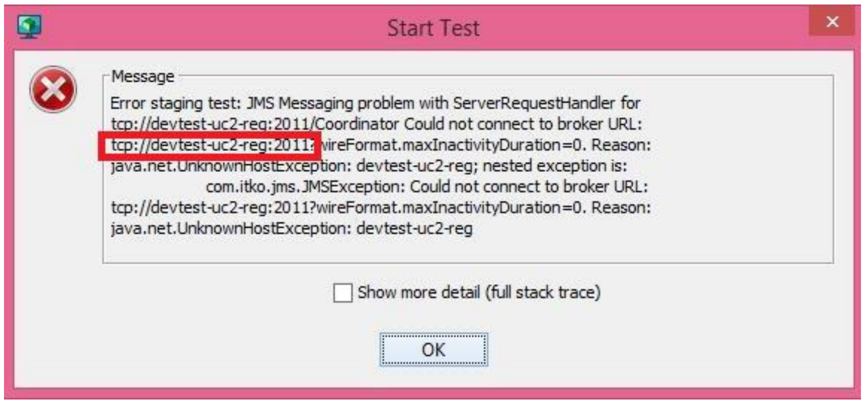
Now we want to test whether or not the kiosk client can connect to virtual service *kioskv6*. This virtual service listens on private port *devtest-u4a-vse:8001*. We have configured a public port on *devtest-archuc2.cloudapp.net:58001* to access this service remotely:

 Kiosk Configuration http:// devtest-archuc2.cloudapp. : 58001 / ... Ok Cancel	Edit the URL to access LISA bank to *devtest-archuc2.cloudapp.net:58001*
 Welcome to LISAfinancial! Error communicating with bank Connection refused RETRY	Click on the Retry button to apply the changed URL …
 Welcome to LISAfinancial! Please select an account CANCEL	… and the kiosk on local workstation is now connected to the virtual LISA Bank application on *devtest-u4a-vse:8001* in Microsoft® Azure. Click on the credit card to select a user, try Lisa Simpson with password golisa, which will log you in. The screenshot on the left shows a successful log in of Lisa Simpson to the virtual service kioskv6.

Stage test from local DevTest workstation to cloud

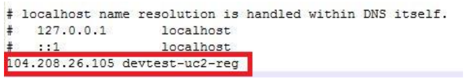
Staging a test case from local workstation to DevTest in Microsoft® Azure requires a name mapping from *devtest-uc2-reg* to *devtest-archuc2-cloudapp.net*.

DevTest workstation retrieves the Coordinator address from Registry service. The Registry service, however, knows the hostname and port number of the Coordinator service that is valid inside the cloud service only. This hostname is not available for remote access, but the cloud service name is. Unless we configure a name mapping from *devtest-uc2-reg* to *devtest-archuc2-cloudapp.net*, we will not be able to access the Coordinator service.



```
Error staging test: JMS Messaging problem with ServerRequestHandler for
tcp://devtest-uc2-reg:2011/Coordinator Could not connect to broker URL:
tcp://devtest-uc2-reg:2011?wireFormat.maxInactivityDuration=0. Reason:
java.net.UnknownHostException: devtest-uc2-reg; nested exception is:
        com.itko.jms.JMSException: Could not connect to broker URL:
tcp://devtest-uc2-reg:2011?wireFormat.maxInactivityDuration=0. Reason:
java.net.UnknownHostException: devtest-uc2-reg
```

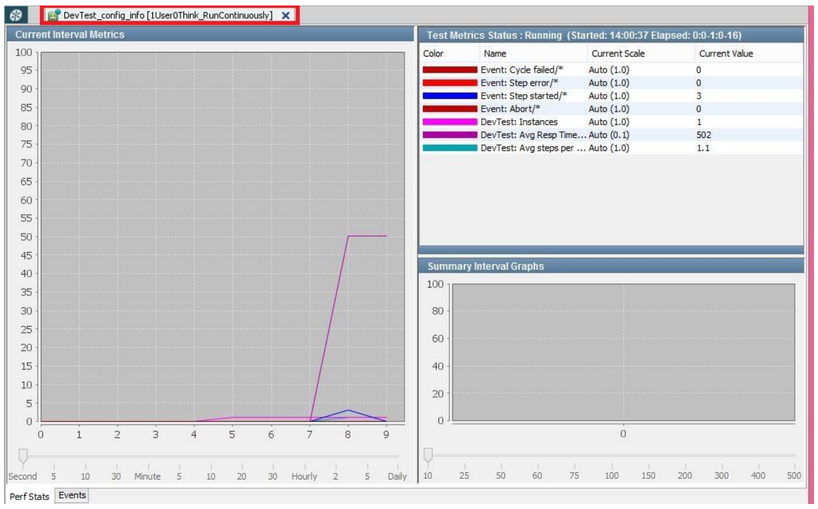
The regular way to map one name to another one is to add a CNAME record to the DNS service. Because we do not have access to the DNS service and as a temporary workaround, we add an entry to the local hosts file, which maps the IP address of *devtest-archuc2-cloudapp.net* to *devtest-uc2-reg*.



```
# localhost name resolution is handled within DNS itself.
#   127.0.0.1       localhost
#   ::1             localhost
104.208.26.105 devtest-uc2-reg
```

This workaround will fail when the Azure cloud service is recycled, because it will then get a new public IP address assigned.

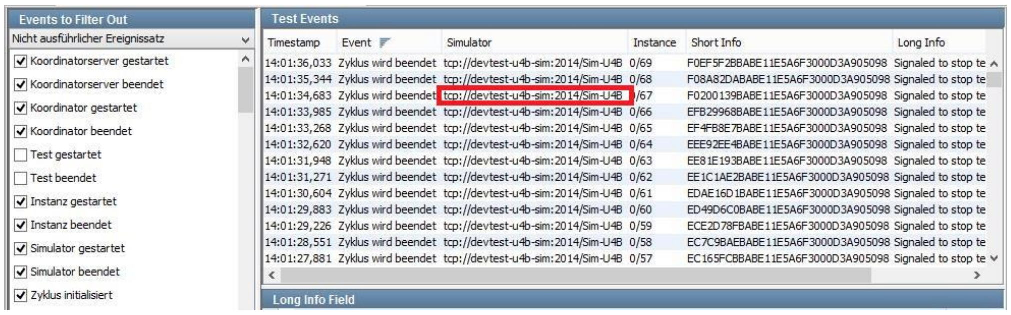
With this workaround, we can now stage a test from local workstation to the Coordinator in Microsoft® Azure, which one of the Simulators in Azure executes. For the screenshots below, we staged the same *DevTest_config_info* test that we used to test the portal.



Test Metrics Status : Running (Started: 14:00:37 Elapsed: 0:0-1:0-16)

Color	Name	Current Scale	Current Value
🟥	Event: Cycle failed/*	Auto (1.0)	0
🟥	Event: Step error/*	Auto (1.0)	0
🟦	Event: Step started/*	Auto (1.0)	3
🟥	Event: Abort/*	Auto (1.0)	0
🟪	DevTest: Instances	Auto (1.0)	1
🟪	DevTest: Avg Resp Time…	Auto (0.1)	502
🟦	DevTest: Avg steps per …	Auto (1.0)	1.1

Watching the test events prove that the test was executed on a Simulator instance in the Microsoft® Azure environment.



Events to Filter Out

Nicht ausführlicher Ereignissatz

- [✓] Koordinatorserver gestartet
- [✓] Koordinatorserver beendet
- [✓] Koordinator gestartet
- [✓] Koordinator beendet
- [ ] Test gestartet
- [ ] Test beendet
- [✓] Instanz gestartet
- [✓] Instanz beendet
- [✓] Simulator gestartet
- [✓] Simulator beendet
- [✓] Zyklus initialisiert

Test Events

Timestamp	Event	Simulator	Instance	Short Info	Long Info
14:01:36,033	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/69	F0EF5F2BBABE11E5A6F3000D3A905098	Signaled to stop te
14:01:35,344	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/68	F08A82DABABE11E5A6F3000D3A905098	Signaled to stop te
14:01:34,683	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	/67	F0200139BABE11E5A6F3000D3A905098	Signaled to stop te
14:01:33,985	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/66	EFB29968BABE11E5A6F3000D3A905098	Signaled to stop te
14:01:33,268	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/65	EF4FB8E7BABE11E5A6F3000D3A905098	Signaled to stop te
14:01:32,620	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/64	EEE92EE4BABE11E5A6F3000D3A905098	Signaled to stop te
14:01:31,948	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/63	EE81E193BABE11E5A6F3000D3A905098	Signaled to stop te
14:01:31,271	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/62	EE1C1AE2BABE11E5A6F3000D3A905098	Signaled to stop te
14:01:30,604	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/61	EDAE16D1BABE11E5A6F3000D3A905098	Signaled to stop te
14:01:29,883	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/60	ED49D6C0BABE11E5A6F3000D3A905098	Signaled to stop te
14:01:29,226	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/59	ECE2D78FBABE11E5A6F3000D3A905098	Signaled to stop te
14:01:28,551	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/58	EC7C9BAEBABE11E5A6F3000D3A905098	Signaled to stop te
14:01:27,881	Zyklus wird beendet	tcp://devtest-u4b-sim:2014/Sim-U4B	0/57	EC165FCBBABE11E5A6F3000D3A905098	Signaled to stop te

Long Info Field

Appendix

VM customization

OS Configuration

- Disable IE Enhanced Security for both administrators and users
- Disable Firewalls for all profiles (private, public, domain)
- Add cmd.exe to taskbar, Start menu and Desktop
- Add permanent share to \\devtest-uc2-reg\c$

Installed Software

- Chrome Browser
- Notepad++
- DevTest Server
    - Pointing to Enterprise Dashboard service on devtest-uc2-reg:2003
    - Installing DevTest services as Windows services without automatic start at start up
    - Demo servers installed

Service configuration

- Updated local.properties according to document
- Reconfigured VSE and Simulator Windows services for automatic start at system start
- Started VSE and Simulator Windows services