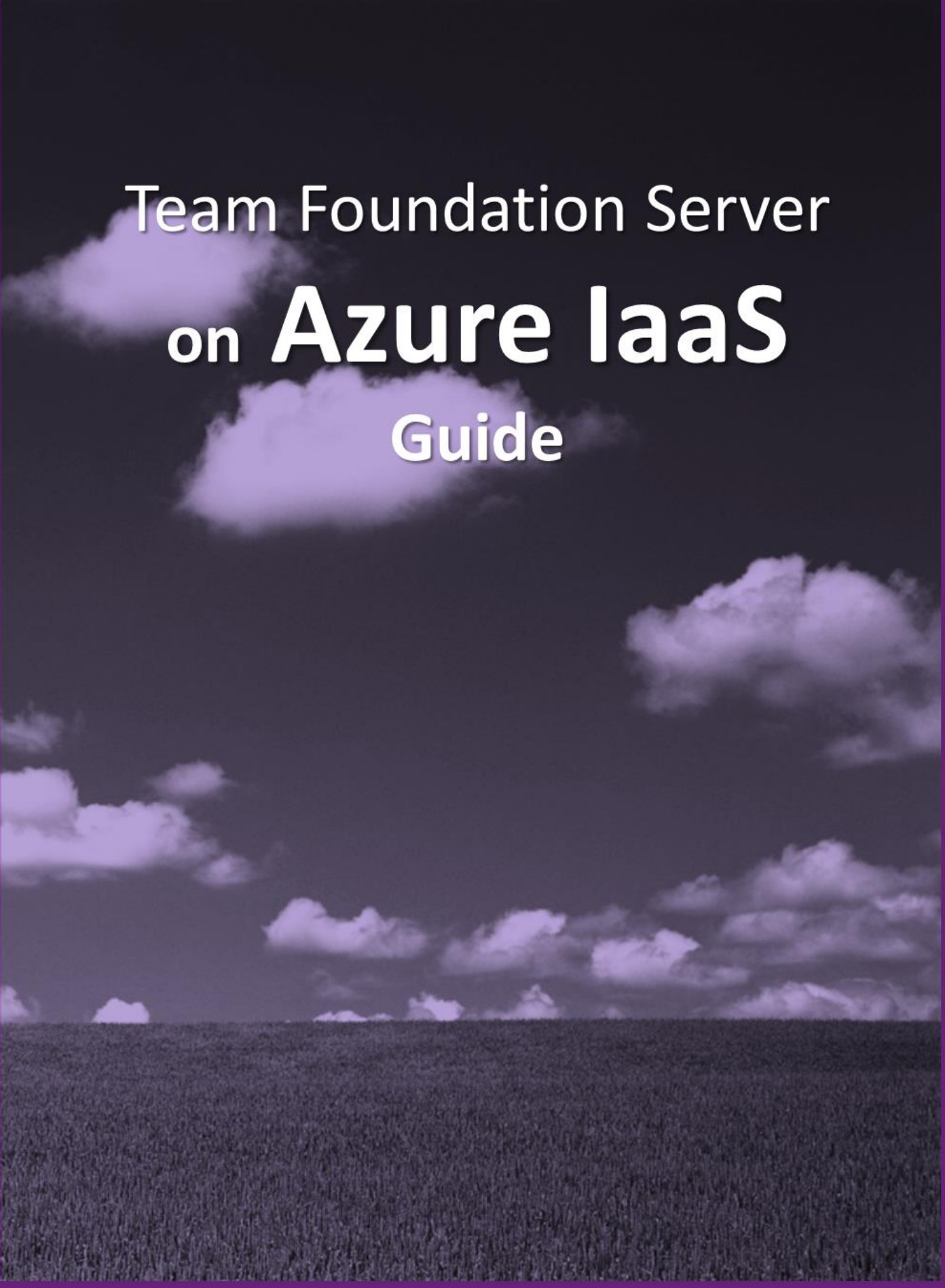


# Team Foundation Server on Azure IaaS Guide

V1.4.2

Visual Studio ALM Rangers

Microsoft

Visual Studio

# Table of Contents

# Foreword

Over the last two years, the industry has seen an incredible adoption of cloud infrastructure to power many of today's IT application workloads. For early adopters the appeal was greater elasticity and resiliency at a lower upfront cost. Today with the maturity of cloud providers, attractive price plans and better on-premises integration IT teams are readying to onboard the vast majority of their application workloads. A shift of budget amounts from CAPEX (capital expenditures) to OPEX (operational expenditures) as a more attractive financial expenditure strategy is further increasing this momentum.

Microsoft's Azure adoption and growth has brought great awareness of the benefits of the cloud. With this awareness, we are seeing more and more teams thinking about adopting a cloud service, like Visual Studio Online, or deploying their ALM application workloads using IaaS (infrastructure as a Service) solutions. Operational control, differences in feature set, and total ownership cost scenarios drive the decision between adopting an online service and going with an IaaS deployment.

Visual Studio Online (VSO) updates the service for, customers with new features every three weeks and "seamlessly" grows to support the organization's load. The service provides Team Foundation Server features like Version Control, Work Item Tracking, Build, and even other ALM workloads like Load Testing. For customers this is great. At the same time, there are some drawbacks around customization and extensibility. For example, currently customers cannot customize the process templates available to team projects, which limits the fields and workflows available to use in the organization. Other drawbacks might include lack of operational control like when updates happen, or datacenter location of the account data.

With an IaaS TFS deployment, customers are able to make full use of the feature set of TFS. Customizations, extensibility, and full operational control are all possible. This version of TFS is the on-premises version, which means TFS always supports moving collections to on-premises. Cloud benefits for elasticity and OPEX costs also apply. There are definitely drawbacks, rate of feature innovation delivered to that instance is significant slower, there are human costs associated with managing the hardware and backups, and setup time can be high.

ALM workloads powered by cloud infrastructures are a reality of the modern IT world. This guide, and the information presented in it by our Rangers team, is a significant first step to allow you to make that transition. We hope it inspires you to run a proof of concept that takes full use of the cloud elasticity, reliability, and OPEX flexibility.

**Mario Rodriguez** *(Senior Program Manager, TFS Infrastructure)*

# Introduction

This document provides practical, scenario-based guidance for the implementation of Team Foundation Server (TFS) on Azure Infrastructure as a Service (IaaS). We guide you through the planning and decisions, based on a real-world proof-of-concept production deployment and experience from the ALM Rangers.

## Intended audience

We expect the majority of our audience personas to be **Dave** – TFS Server Administrator, **Jane** – Infrastructure specialist and **Bill** – ALM Consultant. See ALM Rangers Personas and Customer Types[1] for more information on these and other personas.

The guide assumes a good knowledge of the TFS and an operational administration mindset – in other words, intermediate to advanced TFS Administrators who are looking at the option of moving the TFS environment to the cloud, or ALM Consultants looking for a demo and evaluation environment in the cloud.

## What you'll need

- Microsoft Azure Account
- Team Foundation Server 2013 and prerequisite software

## Visual Studio ALM Rangers

*The Visual Studio ALM Rangers provide professional guidance, practical experience, and gap-filling solutions to the ALM community*. They are a special group with members from the Visual Studio product group, Microsoft Services, Microsoft Most Valuable Professionals (MVPs), and Visual Studio Community Leads. Membership information is available online[2].

## Contributors

Andrea Scripa, Bill Heys, Chris Margraff, Clementino Mendonca , Dan Marzolini, Dave McKinstry, David Pitcher, Eric Golpe, Grant Holliday, Hassan Fadili, Jahangeer Mohammed, James Szubryt, Marcus Fernandez, Mario Rodriguez, Micheal Learned, Oliver Hilgers, Susan Ferrell, Tarun Arora, Tony Feissle, Utkarsh Shigihalli, Willy-Peter Schaub and Wouter de Kort.

## Acknowledgements

ALM Rangers never work alone, standing on the shoulders of giants. We would like to thank our families for their patience, the countless engineers who advised us and shared their knowledge, and content owners from MSDN and TechReady. Special thanks to Brian Harry, Chris Clayton, and Patrick Butler Monterde.

## Additional ALM Rangers Resources

Understanding the ALM Rangers – http://aka.ms/vsarunderstand

Visual Studio ALM Ranger Solutions – http://aka.ms/vsarsolutions

[1] http://aka.ms/treasure4
[2] http://aka.ms/vsarindex

# Concept of the private cloud

GOAL

We introduce the objectives, impact, constraints, challenges and the value-add of a TFS on Azure IaaS environment. This section should enable you to decide whether "in the cloud" is an option for you to consider.

## TFS, Visual Studio Online, and TFS on IaaS

You already know the benefits of TFS for your users from your on-premises deployment. You also know some of the challenges of managing that deployment, and the costs of maintaining it. The cloud offers a cost-effective solution to some of those challenges. Indeed, Microsoft offers its own cloud-based vision of TFS in Visual Studio Online. However, while Visual Studio Online looks similar to TFS on-premises from a user's perspective, customization support is limited, creating a barrier for organizations that have customized processes. Other organizations might have barriers to using a public service, preferring to keep themselves on a private cloud. If you have customized processes, or need integration with other software not currently offered by Visual Studio Online, or just want to explore the possibilities of deploying TFS on a private cloud, read on.

TFS on Azure IaaS refers to a private instance of TFS hosted on Microsoft Azure Infrastructure as a Service. As a private TFS instance, it provides most of the same costs and benefits as Team Foundation Server installed on your hardware within your corporate network. You are in charge of the setup, administration and patching, but also you have complete flexibility in how you plan and configure that instance. For example, you can install with or without SharePoint project portals, utilize a full reporting warehouse, and customize your process templates.

Cloud-hosted infrastructure provides several benefits that are not available in on-premises TFS instances. For example, you can easily scale the instance to add or reduce capacity as needed, you can easily add new capabilities (e.g., new build or proxy servers), and you only pay for what you use.

Many development teams prefer Azure-hosted IaaS servers, as they tend to reduce management overhead and increase development team flexibility.

The objective of this guidance is to enable you to determine whether a complete or partial move to the cloud is beneficial. You will be able to plan, implement, and maintain an effective service in the cloud for product evaluation, testing, training, and production.

NOTE

See Brian Harry's blog: Team Foundation Server on Azure IaaS [3] for information on performance test results, which recommend the TFS on Azure IaaS for less than 1,500 users, to avoid exhaustion of I/O bandwidth for the log and temp db.

## How does TFS on Azure IaaS differ from VS Online?

| | | VS Online | TFS on Azure |
|---|---|---|---|
| **TFS** | Work Items, Source Control and Build features | ☺ | ☺ |
| | Process template customizations | 😐 Not available at the time of writing this guidance. | ☺ |
| | Agile Projects Management | ☺ | ☺ |
| | Test Case Management | ☺ | ☺ |

[3] http://blogs.msdn.com/b/bharry/archive/2013/02/02/team-foundation-server-on-azure-iaas.aspx

| | | VS Online | TFS on Azure |
|---|---|---|---|
| | Heterogeneous Development | | |
| | Near-zero setup and administration | | You must install and administer the operational infrastructure, which incurs cost. Also includes setting up Azure infrastructure and maintenance of the infrastructure. |
| | Collaborate from anywhere | | Azure infrastructure may introduce azure-specific authentication and accessibility constraints. |
| | Virtual Test Lab Management | Not available at the time of writing this guidance. | |
| | SharePoint Integration | Not available at the time of writing this guidance. | |
| | Data Warehouse & Reporting | Not available at the time of writing this guidance. | |
| **Environmental** | Does not need infrastructure to host TFS. | | TFS on Azure may require additional time to setup the infrastructure. |
| | Zero environment setup and maintenance | | |
| | On-premises infrastructure with unpredictable usage spikes | | TFS on Azure infrastructure needs to cater for extra scalability and fault tolerance to accommodate the unpredictable spikes. This investment may not meet ROI criteria. |
| | Can accommodate anticipated usage that exceeds deployment's infrastructure capacity | | TFS on Azure infrastructure needs to allow scalability without downtime, which is a costly investment that may not meet ROI criteria. |
| | Need for total control of infrastructure for auditing | | |
| | Multiple partner companies needing an integrated solution | | TFS on Azure infrastructure may introduce premise-specific authentication and accessibility constraints. |

| | | VS Online | TFS on Azure |
|---|---|---|---|
| | Data in motion / at rest constraints for compliance purposes (HIPAA, PCI, etc.), such as the strictly regulated gaming industry | ☹ | ☺ |
| **Teams** | Small <250 | ☺ | ☺ |
| | Medium <500 | ☺ | ☺ |
| | Large 500+ | ☺ | ☺ |
| | Geographically distributed | ☹ No choice of geographic location (data stored in data center in Chicago [4]) at the time of writing this guidance | ☺ |

**Table 1 – Comparing VS Online with TFS on Azure IaaS**

**Key: ☺ = yes, 😐 = partial, ☹ = no**

[4] http://blogs.msdn.com/b/bharry/archive/2014/02/10/visual-studio-online-update-feb-10.aspx

# Azure concepts

To plan your TFS on Azure IaaS environment, you will need to understand the key considerations and associated guidance for Azure. This section elaborates key concepts summarized in **Table 1 – Comparing VS Online with TFS on Azure IaaS**, page 9, which we document in more detail on the Microsoft Azure [5] website.

## Azure Subscription

### Selecting a Azure Subscription

To deploy TFS on Azure IaaS, you will need a Microsoft Azure account to hold the TFS instance. You will want to plan for that cost. The following table summarizes the different types:

| Type | Description | Note |
|---|---|---|
| Pre-paid | Suitable if you already have a corporate commitment for Azure. You should consider if you already know your organization will use Azure. Following your evaluation, you can convert a paid account into a full production TFS instance. | Refer to purchase options [6] or contact your Microsoft account manager to setup a prepaid account. |
| Pay-as-you-go | Offers the most flexibility and no long-term commitment. As with a prepaid account, you can convert a TFS installation configured under this account into a full production instance following the evaluation period. | |
| Trial | Can be used with no upfront commitment or expense. Microsoft offers a one-month trial including $200 USD of usage credit. Following the trial, you can convert the account to a full production Pay-As-You-Go or prepaid commitment account. | A trial subscription can be obtained at free trial [7] |
| MSDN | Provides between a $50 and $150 monthly benefit for Microsoft Azure, depending upon the level of the MSDN subscription. For the first month after activating the Azure benefit, MSDN accounts receive $200 in usage credit, the same as a trial subscription. In addition, you can optionally associate MSDN subscriptions to a credit card to allow the best-discounted rates for Azure usage beyond the monthly benefit. Microsoft Azure benefits associated with MSDN subscriptions can only be used development and test purposes, such the TFS evaluation configuration described in this guide. However, you can transfer any assets generated during the evaluation to a production-use Microsoft Azure subscription to expedite a production deployment. | Refer to subscription benefits [8] for information on MSDN subscription benefits and benefits [9] for information on activating your Microsoft Azure MSDN benefits. |
| Internal | Limited to Microsoft internal use. | Used for the companion proof of concept |

**Table 2 - Azure subscription types**

[5] http://www.windowsazure.com/

[6] http://www.windowsazure.com/en-us/pricing/purchase-options

[7] http://www.windowsazure.com/en-us/pricing/free-trial

[8] http://msdn.microsoft.com/subscriptions/aa718661

[9] http://www.windowsazure.com/en-us/pricing/member-offers/msdn-benefits

## Accessing your Microsoft Azure Subscription

In order to install and configure a TFS proof-of-concept environment on Azure, you will need to understand how to use Azure. Although you will perform some of the configuration by logging into the servers running in Azure, some of the configuration will entail configuring the Azure subscription.

There are two ways to manipulate Microsoft Azure: **programmatically** and **manually**.

- When manually interacting with Microsoft Azure, use a web browser and the Azure management portal. To access the Microsoft Azure management portal, direct your browser to http://manage.windowsazure.com and log in with the account associated with your subscription.

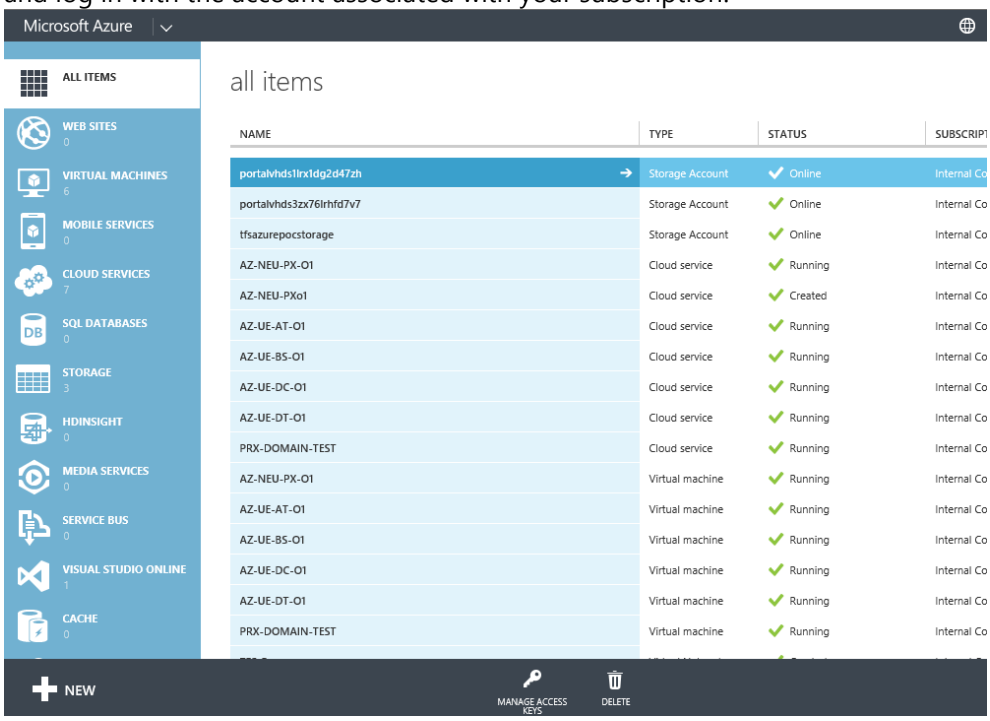



**Figure 1 – The Microsoft Azure management portal.**

This figure depicts a snapshot of the management portal. Your portal might not look like this. The Azure team continues to evolve and improve all Azure tools, so the portal will evolve in terms of appearance and functionality.

- There are several ways to manipulate programmatically Azure.
  - REST APIs allow you to write code using your favorite development technologies.
  - The cross-platform command line interface, based on NodeJS, allows you to script and manipulate Azure environments.
  - A full list of the .NET and REST API functionality is available through the Microsoft Azure Reference[10].

We chose the manual interaction for this guide to allow us to walk-through the installation and configuration steps. We do not cover any of the programmatic options.

# Azure Platform Governance

Before you go any further, you need to define the governance for billing, security, licensing and other corporate policies, such as:

- **Billing** - Do you need to bill per project, group division, corporate?
- **Security** – Do you need to follow any corporate policies? Who is responsible for the Azure subscription?

[10] http://azure.microsoft.com/en-us/develop/net/reference/

- **Licensing** – Do you need to control indiscriminate installation of software?
- **ALM** – Are there any policies around shared services, version control, and work item management?

The Azure Platform provides services grouped in a logical unit called a subscription. Small companies might choose individual subscriptions for project members, such as a single account, owned by one Account Owner, and a single internal subscription. Companies with an enterprise license will have a higher level of management, with an ability to create multiple subscriptions as shown below:

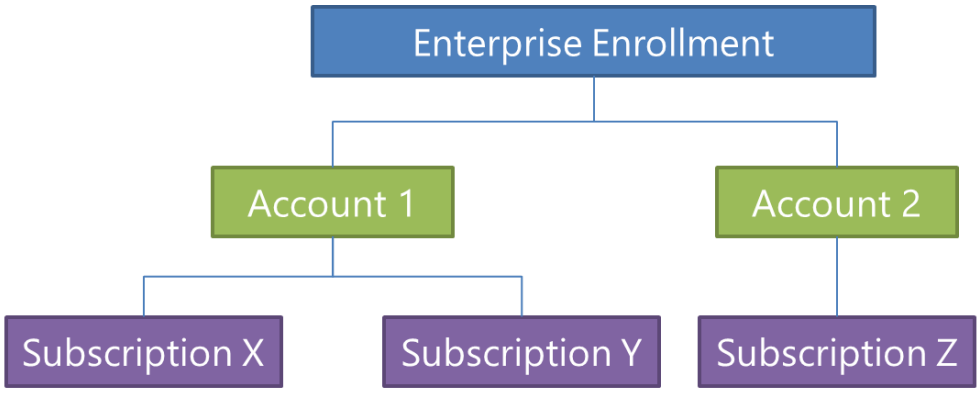



**Figure 2 - Azure Platform Governance**

For more information, see Manage Accounts, Subscriptions, and Administrative Roles [11] on the Azure documentation portal.

We do not base the proof-of-concept included with this guidance on an Enterprise license. Instead, it uses a single account, owned by one Account Owner, and a single internal subscription. It simulates the common account and subscription taxonomy.

# Azure Network

Next, you need to understand the Achilles heel of any successful TFS on Azure IaaS environment: the network. See the TFS on Azure IaaS is easy … if you have a healthy network [12] video, which introduces this guidance and walks through the proof-of-concept network in-depth. This section will explore a quick overview of the following network artifacts:

- Access and authentication
- Network topology
- Affinity groups
- Address segments
- Name resolution

## Security and Authentication

You need to plan how you authenticate users who access TFS on Azure IaaS.

The most common TFS deployment runs in an Active Directory domain, and users log in with domain credentials.

There are three methods for implementing an Active Directory domain in your Azure subscription:

- Configure Azure Active Directory to maintain a separate directory instance in the Azure environment.
- Use Azure Active Directory Access Control to use your on-premises domain logins in the Azure environment.

[11] http://msdn.microsoft.com/en-us/library/windowsazure/hh531793.aspx

[12] http://aka.ms/vsarPlanningAzureVideo

- Configure domain controllers by running Windows Server virtual machines promoted into the Azure domain role.

The proof-of-concept assumes that you will have an on-premises domain responsible for authentication all users and a separate Active Directory in Azure, which will contain and secure the Azure assets.

## Security Considerations

When choosing a method to authenticate users in your environment, consider Azure as another data center into which you have extended your environment. Configure servers in a management domain that is separate from user logins to give an extra measure of protection if one of your front-end servers is compromised. While you can accomplish this with any one of the three methods, using Azure Active Directory has no dependencies on your existing environment, and you will be able to use this domain across other applications in your Azure subscriptions. You will be able to connect your Azure domain to Microsoft Azure Active Directory using a one-way trust, where the Microsoft Azure Active Directory trusts logins from your parent corporate directory.

For more information, see Guidelines for Deploying Windows Server Active Directory on Microsoft Azure Virtual Machines [13].

You can accomplish these same goals using Azure Access Control. Azure Access Control allows you to leverage external authentication methods, such as Microsoft or Facebook accounts, which may be well suited for smaller teams.

For information about configuring Microsoft Azure Access Control, see the article Access Control Service 2.0 [14].

## Planning your Domain

If you are going to access your TFS environment from the internet, you will need to configure the appropriate DNS records to point to your front-end environment. If an ISP hosts your corporate domain, you will need to consult the ISP hosting your domain to complete this task. If you are using Active Directory or an on premises DNS to manage your domain or a subdomain of an internet published domain, you can join machines to your corporate domain and use AD-integrated DNS for name resolution.

## Planning Authentication

There are several options available for identity management. TFS requires that users be able to log in using NTLM or Kerberos, and service accounts in a multiple-server TFS farm must be domain accounts. Choose from the options in the table below when configuring the environment.

| Method | Environment Size | Details | Pros | Cons |
|---|---|---|---|---|
| **Standalone Workgroup (No domain)** | Not Scalable. Singe Server. | TFS installed as a workgroup server with Identity management supplied by local user accounts. | Very low administrative overhead. Users and suppliers have an equal access method, | Corporate Users do not get a single-sign on experience. Environment not scalable into a larger farm. |
| **Standalone domain** | Scalable | A separate AD instance, completely independent of any other domain. | Users and suppliers have an equal access method. Virtual network not needed for authentication purposes. | Corporate Users do not get a single-sign on experience. |

[13] http://msdn.microsoft.com/library/windowsazure/jj156090.aspx

[14] http://msdn.microsoft.com/library/hh147631.aspx

| | | | | |
|---|---|---|---|---|
| **Extend domain** | Scalable | Corporate domain extended into the Azure environment by deploying domain controllers or as a new Site. | Corporate users have a single-sign on experience | External users would need an account in the corporate domain and have possible access to resources inside the corporate environment. |
| **One Way Trust** | Scalable | Define a new domain in Azure, with a one-way trust to the SFA domain. Suppliers would have accounts in the new domain only. | Corporate users have a single-sign on experience. External users prevented from accessing other resources inside the company firewall. | External users could gain access to resources inside the company firewall if network segments are exposed other than AD/DNS. |

**Table 3 – Planning authentication**

Consider the security and management implications of extending your AD domain into Azure. A best practice used in our proof of concept was to establish a ***one-way trust*** between an existing on-premises AD and a new AD forest hosted on Azure Virtual Machines. This allows creating Azure machine using the Azure domain while the corporate directory provides user authentications. Review the standard guidance for configuring a one-way trust with an external domain, which you can find in the article: [How Domain and Forest Trusts Work: Domain and Forest Trusts](#) [15].

Performance of authentications going across the tunnel to the corporate AD has not been a bottleneck in our proof-of-concept tests. If your network link is slow or unstable, you may want to consider placing a read-only domain controller in the Azure environment to move authentication traffic off the VPN tunnel.

### Load Balancing

Each cloud service you set up will have its own unique web address. By default, each VM will have its own cloud service. You can implement load balancing between front-end nodes by configuring Azure Traffic Manager. For more information, see the article [Microsoft Azure Traffic Manager](#) [16].

### Create Microsoft Azure Active Directory

Typically, you use Microsoft Azure Management Portal to manage the services associated with your Microsoft Azure subscription. One of the newer Azure services that you can use for identity management and directory tenant capabilities is the Active Directory service. If you are an administrator, you can manage these capabilities by clicking on Active Directory in the left-navigation of the Management Portal.

### Domain Joining Virtual Machines

You will want to join your virtual machines to the local domain in the Azure environment.

## Network Topology

Once we have a clear picture of the security and authentication requirements, we can plan and configure the network topology. In order to connect your Azure IaaS TFS environment to resources on your local network, you will need to configure a network topology to connect your local resources to the TFS in Azure IaaS resources.

[15] http://technet.microsoft.com/en-us/library/cc773178(v=WS.10).aspx

[16] http://azure.microsoft.com/en-us/services/traffic-manager/

Azure virtual networking allows you to connect your Azure subscription to your local network in much the same way you would connect a remote office or secondary data center.

You will need to perform the following steps to configure the network:

- Plan network address segments
- Plan affinity grouping
- Plan name resolution
- Setup Virtual Network [17]

We cover these steps in the proof-of-concept walkthroughs. Let us take a quick tour through of network address segments, affinity grouping, name resolution, and tunneling to set the scene. See Azure Networking [18] for information that is more detailed.

## Network Address Segments

To create the Azure virtual network, you need to determine the address space to use in the subscription, and then create separate network segments of this address range for:

- **A Front-End subnet**, which will receive all incoming HTTP requests for all services configured in the cloud.
- **A Back-end subnet**, which will be visible to the front-end devices but will be blocked from incoming HTTP traffic.
- **A Gateway subnet**, which will provide IP address space for gateway functions that connect your Azure subscription to on-premises resources.

The network address segment you choose:

- Must be unique on your LAN
- Must be an unused address space on your LAN
- Must be non-routable per RFC 1918 [19]
- Must contain enough IP addresses for all resources you will add to the subscription.

## Affinity Grouping

An affinity group is a logical component that groups Microsoft Azure Services together in an Azure datacenter. When you are using the virtual network creation wizard, TFS will ask you to assign an affinity group to your virtual network. You must use an affinity group to assure that your TFS deployment has **minimal latency** between servers in the environment. Learn more about affinity groups in About Affinity Groups for Virtual Networks[20].

Give the affinity group a name describing its application (e.g. "TFSServicesGroup") and decide which Azure data center is closest to the majority of your users. Once you build your environment in this affinity group, you **cannot** move it and you will need to start your deployment over to change it.

See Create an Affinity Group in the Management Portal [21] for more information.

## Name Resolution

Your name resolution approach may vary depending on how you configure identity management for your deployment. We suggest you create a local Active Directory in Azure and establish a one-way trust with your on-premises Active Directory. However, for your initial deployment, consider using existing on-premises DNS servers

[17] http://msdn.microsoft.com/en-us/library/windowsazure/jj156007.aspx

[18] http://msdn.microsoft.com/en-us/library/azure/gg433091.aspx

[19] http://tools.ietf.org/html/rfc1918

[20] http://msdn.microsoft.com/en-US/library/windowsazure/jj156085.aspx

[21] http://msdn.microsoft.com/en-US/library/windowsazure/jj156209.aspx

for name resolution until you have successfully deployed the virtual network. You will be able to change your DNS configuration after deploying the Azure Virtual Network.

Note that during configuration of the virtual network you are not required to enter a DNS Server. However, for this project it is essential that you do so. Azure will use default DNS servers for name resolution that will not allow you to resolve on-premises computers from Azure. For more information read the article Name resolution using your own DNS server [22].

# VPN Tunnel

## About VPN Tunneling

A VPN tunnel allows the IP address range configured for Azure to be visible to and from your local network. This tunnel is an encrypted connection that you configure on the customer premises and in your Azure virtual network. Azure offers both point-to-site and site-to-site configurations, and supports both static and dynamic routing.

## Establish the VPN tunnel

Azure virtual networking configuration defines the address space that resources deployed in a subscription can utilize. Just as you would establish an IP address space for a local area network and use a gateway to connect your network to the internet, the Azure Virtual network does the same for your Virtual Machines in your cloud subscription. Note that you must use non-routable IP address ranges (10.x, 172.x, 192.x) as defined by IETF RFC 1918 [23] for virtual networking configuration.

See Microsoft Azure Virtual Network Overview [24] and Create a Virtual Network in Microsoft Azure [25] more information.

The Azure virtual network does not connect your Azure environment to your local network. The network configuration walkthrough, page 37, will assist with setup of a VPN tunnel for this purpose.

## Tunneling scenarios

- **Point-To-Site**, which uses a software agent installed on the on-premises side to create the gateway for the VPN to Azure.
- **Site-To-Site**, using a VPN device (i.e. Cisco, Juniper)
- **Site-To-Site**, which uses Routing and Remote Access Service (RRAS) on Windows Server 2012. The RRAS route is a pure software based Windows Server 2012 Site-to-Site VPN option, which we can activate by running a PowerShell script from the Microsoft Azure Management Portal that enables Routing and Remote Access Service (RRAS) on the Windows Server and configures a Site-to-Site VPN tunnel and routing table on it. This is the scenario used by the companion proof-of-concept.

The key difference is the consideration of usability of VPN and DNS on a site-level (subnets), or by individual machines (or groups of machines). For example, if you want to enable branch office or dedicated VPN scenarios, you will need to go with Site-To-Site with dedicated hardware, versus Point-to-Site.

[22] http://msdn.microsoft.com/en-US/library/windowsazure/jj156088.aspx#bkmk_BYODNS

[23] http://tools.ietf.org/html/rfc1918

[24] http://msdn.microsoft.com/en-us/library/windowsazure/jj156007.aspx

[25] http://www.windowsazure.com/en-us/manage/services/networking/create-a-virtual-network/

The online guidance is as follows:

| Scenario | Points to consider | For more information … |
|---|---|---|
| Secure site-to-site connection between your virtual network and your on-premises network | • Address space<br>• Supported VPN gateway device<br>• Internet-accessible IP address for your VPN gateway device<br>• Name resolution (DNS) design | See About Secure Cross-Premises Connectivity [26] for more information about cross-premises connection options.<br>See About VPN Devices for Virtual Network [27] for VPN device requirements and configuration templates. |
| Secure point-to-site connections between individual computers running on your on-premises network and your virtual network | • Address space<br>• Name resolution (DNS) design | |

**Table 4 - Tunneling considerations**

## Tunneling services to consider

### Multiprotocol Label Switching (MPLS)

**With Microsoft Azure ExpressRoute**, you can create private, high-throughput connections between Azure datacenters and your existing infrastructure, whether it is on-premises or in a colocation environment. ExpressRoute connections do not go over the public Internet, and they offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet.

For more information, visit the ExpressRoute [28] website. To get a comprehensive look at pricing, visit the ExpressRoute Pricing Details [29] website.

### vnet-to-vnet (wide vnet) and inter-vnet (multisite)

**Virtual Network** now supports more than one site-to-site connection, allowing you to connect securely multiple on-premises locations with a virtual network (VNET) in Azure.

**VNET-to-VNET connectivity is enabled**, allowing multiple virtual networks to connect directly and securely with one another. Using this feature, you can connect VNETs that are running in different Azure regions and have traffic route via the Azure backbone.

See Virtual Network [30] website for details.

# Global TFS deployment

One of the strongest business cases for deploying TFS to Azure IaaS is its ability to adapt rapidly to a global development effort. Modern development teams might include members from around the globe, all working in different time zones and potentially making conflicting updates to source code.

Team Foundation Server is an excellent tool to manage source code conflicts and implement strong test and validation approaches.

[26] http://msdn.microsoft.com/en-US/library/azure/dn133798.aspx

[27] http://msdn.microsoft.com/en-US/library/azure/jj156075.aspx

[28] http://azure.microsoft.com/en-us/services/expressroute/

[29] http://azure.microsoft.com/en-us/pricing/details/expressroute/

[30] http://azure.microsoft.com/en-us/services/virtual-network/

Team Foundation Server Proxy provides helps improve performance for the remote team. In the Azure environment, you can configure a TFS Proxy and make it operational within minutes to support a remote team.

# Virtual Machines

WARNING

If there is a flaw in your virtual network configuration, in many cases you will need to rebuild completely the virtual network to proceed. This can mean deleting and rebuilding your virtual machines. Be sure you have rigorously tested your virtual network and access configurations before proceeding to creating VMs and building your TFS environment.

## Planning Environment

We recommend that you use the TFS Planning Guide, which is part of this TFS Planning and DR Avoidance [31] guidance and the Hardware Recommendations [32] to plan your TFS environment. Use the virtualization guidelines to define the virtual machines "as if" you were planning an on-premises VM based environment.

For the data tier (DT) and the planning of the SQL Server we further recommend SQL Server in Microsoft Azure Virtual Machines [33] and Performance Guidance for SQL Server in Microsoft Azure Virtual Machines [34] for detailed guidelines.

You should become familiar with the concepts provided in Create a virtual machine running Windows Server [35] before implementing virtual machines in Azure.

## Sizing and mapping VMs

There is no 1:1 mapping between on-premises VMs and Azure VMs. The following table gives a rough mapping of the scenarios covered in the TFS Planning and DR Avoidance Guide with the Virtual Machine and Cloud Service Sizes for Azure [36].

| Scenario | Example specification | Users | Basic, Standard Azure VM estimate |
|---|---|---|---|
| **Single Server** | 1 x ATDT: 1 Processor, 2GB RAM, 125GB Disk | 250 | A1 |
| | 1 x ATDT: 1 Dual Core Processor , 4GB RAM, 300GB Disk | 500 | A2 |
| **Scale-up Servers** | 1 x AT: 1 Dual Processor, 4GB RAM, 500GB Disk<br>1 x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk | 2200 | A2<br>A3 |
| | 1 x AT: 1 Quad Core Processor , 8GB RAM, 500GB Disk<br>1 x DT: 2 Quad Core Processor, 16GB RAM, 3TB Disk | 3600 | A3<br>A4 |
| **Scale-out Servers** | n x AT: 1 Dual Processor, 4GB RAM, 500GB Disk<br>m x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk | 3600+ | A2<br>A3 |

**Table 3 – Maximum recommended users matrix compared to Azure VM Sizes**

[31] http://aka.ms/treasure5

[32] http://msdn.microsoft.com/en-us/library/dd578592.aspx

[33] http://msdn.microsoft.com/en-us/library/windowsazure/jj823132.aspx

[34] http://msdn.microsoft.com/en-us/library/windowsazure/dn248436.aspx

[35] http://www.windowsazure.com/en-us/documentation/articles/virtual-machines-windows-tutorial/

[36] http://msdn.microsoft.com/en-us/library/azure/dn197896.aspx.

## TFS farm in Azure IaaS

After you have configured a virtual network, determined and implemented your Active Directory and security strategy, you are ready to create your TFS farm. Building machines for TFS in Azure IaaS is the same as it would be if you were building a TFS farm on premises. See How to: Create a Team Foundation Server farm (high availability) [37] for more information.

## Plan VM environment

Before you start the creation of VMs, plan the following:

- **Subscription Administrator(s)**
  Users who are responsible for building and configuring the VMs.
- **List of machine(s)**
  List of all virtual machines you plan to build and support in your TFS environment, including naming and designating its size. See **Checklists**, page 23, for examples.
- **Naming convention**
  List of virtual machines that cancan be easily identified in the Azure portal. See **POC naming conventions**, page 32, for examples of our companion proof-of-concept VMs.
- **Endpoints**
  An endpoint is a TCP or UDP port that you want to open to traffic on your virtual machines. You will generally configure endpoints to allow traffic on your front-end subnet. You can also enable other endpoints for your other subnets. You can configure endpoints after the virtual machine has been enabled. Think of an endpoint as a firewall mechanism. By default all ports are blocked access to your virtual machine (except for RDP) until you configure an endpoint to allow traffic to access the virtual machine.
- **Load Balancing (and Endpoints)**
  If you configure multiple front-end servers, you will want to load balance these servers using a load balancer. Azure supports only **round-robin** load balancing at this time. To configure load balancing, you create a new endpoint and assign it to one of your virtual machines. On the next and all subsequent virtual machines, you will choose **Add Endpoint**, then select the option to **Load Balance Traffic on existing endpoint**, and choose the endpoint you created for the first virtual machine. See Configure Round Robin Load Balancing [38] for more information.
- **Storage accounts**
  VHDs you upload or create will be hosted in an Azure Storage account. By default, a new Azure Storage account will be created for each VM that you create, and your VHDs will be stored in the \VHDS folder on this storage account. You will want to create a single storage account to contain all your virtual disks or you will soon have a large, unmanageable collection of storage accounts with cryptic names, and little ability to find VHDs when you need to. See **Storage**, page 20, for more details.

## VM Maintenance

The walkthroughs cover the creation of the virtual machines. Here are a few VM maintenance tasks to consider.

### Domain Join

Join each of your virtual machines to the domain. Your Active Directory instance will take care of registering your machine name with your DNS, and you will be able to use domain accounts in your machine configurations.

Once you join your virtual machine to a domain and restart it, record the IP address assigned to the virtual machine. Azure's DHCP service will assign an IP address for the pool available in the subnet in which you are

[37] http://msdn.microsoft.com/en-us/library/ee259689.aspx

[38] http://msdn.microsoft.com/en-us/library/ee259689.aspx

building the machine. This address will not change for the life of the virtual machine. You can shut down and restart the virtual machine without losing this address. If you de-allocate the virtual machine and build a new one from its VHD at some point, Azure will assign a new IP address.

### Upload a virtual machine

If you plan to upload a virtual machine, you can use the CSUPLOAD utility provided in the Microsoft Azure SDK to upload from the command line, or you can use an Azure storage utility. You can find a list of Azure storage utilities in Microsoft Azure Storage Explorers [39]. We evaluated the Azure Storage Explorer [40] from Neudesic [41] during the creation of this guidance and performed flawlessly.

After you upload your Virtual Hard Drive, follow the guidance for creating a VM as noted above. Instead of picking a default image from the gallery, choose "From VHD," and select the VHD you have just uploaded.

### Using the Azure Image gallery

You can take a great deal of work out of building the same machine images over and over by uploading a virtual machine and placing it in your image gallery. After you have uploaded a virtual machine, select the "Images" view on the Virtual machine tab and select **New**. You must name the virtual machine and ensure that you have ran the SYSPREP utility before creating an image from the VHD. After completing this dialog, your virtual machine will appear in the **Form Image** section when viewing the Gallery.

## Storage

Lastly, consider storage options and constraints in the Azure environment as VMs and TFS can consume considerable storage space. Azure Blob storage is a service for storing large amounts of unstructured data that can be accessed from anywhere in the world via HTTP or HTTPS, used for images and fixed sized disks in the context of TFS on Azure IaaS.

| Storage Account | Guideline / Limit |
|---|---|
| Name | Name your storage account with something specific to your service, such as TFS_Disks. Each time you create a virtual machine, you will want to designate this storage account as the target for the related Files. |
| Capacity | Up to 200TB |
| Performance | Single blob can handle up to 60Mbytes/sec |
| Transactions | Up to 20,000 entities/messages/blobs per second |
| Geo Redundancy [42] | • Ingress - up to 5 gigabits per second<br>• Egress - up to 10 gigabits per second |
| Locally Redundant [43] | • Ingress - up to 10 gigabits per second<br>• Egress - up to 15 gigabits per second |

**Table 3 – Azure storage guidelines**

[39] http://blogs.msdn.com/b/windowsazurestorage/archive/2010/04/17/windows-azure-storage-explorers.aspx

[40] http://azurestorageexplorer.codeplex.com

[41] http://www.neudesic.com/

[42] http://blogs.msdn.com/b/windowsazure/archive/2012/06/08/introducing-locally-redundant-storage-for-windows-azure-storage.aspx

[43] http://blogs.msdn.com/b/windowsazure/archive/2012/06/08/introducing-locally-redundant-storage-for-windows-azure-storage.aspx

See How to use the Microsoft Azure Blob Storage Service in .NET [44] and Microsoft Azure Storage Scalability and Performance Targets [45] for latest details about storage account capacity and for more information on Azure storage.

Now that you have an overview of the cloud and Azure, you can start planning your own TFS on Azure IaaS environment.

NOTE

Based on findings from the proof-of-concept and real-world environments we recommend you:

- Splitting things onto dedicated disks and adding IO.
- Be prepared to optimize and go heavy on storage.

If you have performance issues, it will likely be there.

# Performance Learnings

NOTE

This is a volatile section. We will revise the content as we continue to learn from our in-the-field proof-of-concepts and production environments.

## Application Tier

- Add a TFS Proxy server to reduce the load on the AT server.

## Data Tier

- Allocate SQL server memory … lots!
- Consider splitting MDF, LDF and TempDB on separate Azure disks.
- Consider striped volume for data.
- Consider using SQL Server 8032 Startup Trace Flag
- Using SQL Server in Microsoft Azure Virtual Machine? Then you need to read this… [46].
- Using Storage Spaces on an Azure VM cluster for SQL Server storage [47]

# Out-of-scope Reference Information

NOTE

We are not covering all scenarios in this guidance, planning to fill the gaps over time, as we gather experience in real-world scenarios. In this section you will find reference information to other sources for some of the out-of-scope scenarios.

## Data Transfer

Ship-a-disk data transfer mostly suited to very large transfers. See Import/Export Pricing Details [48] for details.

[44] http://www.windowsazure.com/en-us/develop/net/how-to-guides/blob-storage/

[45] http://msdn.microsoft.com/en-us/library/dn249410.aspx

[46] http://blogs.msdn.com/b/psssql/archive/2014/06/12/using-sql-server-in-windows-azure-virtual-machine-then-you-need-to-read-this.aspx

[47] http://blogs.msdn.com/b/dfurman/archive/2014/04/27/using-storage-spaces-on-an-azure-vm-cluster-for-sql-server-storage.aspx

[48] http://azure.microsoft.com/en-us/pricing/details/storage-import-export/

## Lab Management

Integration available with TFS as a service. See Continuous delivery to Azure using Visual Studio Online [49] and Continuous Delivery for Cloud Services in Azure [50] and from one of our MVPs Using Visual Studio 2012 Lab Manager to Create a Build Lab in the Sky [51].

[49] http://www.windowsazure.com/en-us/develop/net/common-tasks/publishing-with-tfs/

[50] http://www.windowsazure.com/en-us/develop/net/common-tasks/continuous-delivery/

[51] http://blogs.msmvps.com/molausson/2012/07/20/using-visual-studio-2012-lab-manager-to-create-a-build-lab-in-the-sky

# Checklists

## Planning

The planning steps checklist guides you through the TFS Azure on IaaS planning process.

| Step | Instructions |
|---|---|
| 1<br>Determine Azure IaaS suitability<br>☐ - Done | Cloud computing delivers computing capabilities as a service, giving you access to resources like compute power, networking, and storage. Decide whether the constraints are acceptable and that there is a value-add.<br>• Review section **Concept of the private cloud**, page 7.<br>• Review Microsoft Azure [52] and What is Cloud Computing [53] for more information. |
| 2<br>TFS planning<br>☐ - Done | Use the TFS Planning and DR Avoidance Guide [54] to determine the best server architecture for your requirements.<br>• Review section **Plan the TFS environment**, page 30.<br>• Review Team Foundation Server Architecture [55] for more information. |
| 3<br>Azure IaaS mapping<br>☐ - Done | There is no perfect mapping between Azure VMs and on-premises machines. Map to the closest Azure VM configuration, based on cost and scalability.<br>• Review section **Sizing and mapping VMs**, page 18.<br>• Review section **Virtual Machines**, page 18.<br>• Calculate the estimated cost using the Azure Pricing Calculator [56].<br>• Define your POC architecture. See POC Architecture "at a glance" page 33 for an example of our POC architecture. |
| 4<br>Azure Platform Governance<br>☐ - Done | Define the account, subscriptions, and ownership of administration and reporting to meet your enterprise auditing requirements. Clearly define ownership and responsibilities.<br>• Review section **Azure Subscription**, page 10.<br>• Review section **Azure Platform Governance**, page 11. |
| 5<br>Network planning<br>☐ - Done | Define your network, which encompasses on-premises requirements, network address segments, affinity grouping, name resolution, Azure tunneling, and monitoring.<br>• Review section **Azure Network**, page 12.<br>• See Deploying TFS completely in the Cloud [57] if you are looking for an Azure only environment.<br>• Meet with your on-premises networking and administration team to understand the constraints and the options. You will need to talk to your network, hardware, and security administrators.<br>• Define your naming conventions to maintain consistency and simplicity. See page 32 for our proof-of-concept naming guidelines. |
| 6<br>Storage planning | Define your storage strategy, using on-premises storage or Azure storage.<br>• Review section **Storage**, page 20.<br>• Review Using SQL Server in Microsoft Azure Virtual Machine? Then you need to read this. [58] |

[52] http://www.windowsazure.com

[53] http://www.windowsazure.com/en-us/overview/what-is-cloud

[54] http://aka.ms/treasure5

[55] http://msdn.microsoft.com/en-us/library/ms252473.aspx

[56] http://www.windowsazure.com/en-us/pricing/calculator

[57] http://blogs.msdn.com/b/chrismargraff/archive/2014/05/20/deploying-tfs-in-the-cloud.aspx

[58] http://blogs.msdn.com/b/psssql/archive/2014/06/12/using-sql-server-in-windows-azure-virtual-machine-then-you-need-to-read-this.aspx

| | |
|---|---|
| ☐ - Done | • Review Using Storage Spaces on an Azure VM cluster for SQL Server storage [59] |
| 7<br>Migration and backup planning<br>☐ - Done | Define your data migration, backup and restore strategy. Here are a few useful references. Note that SharePoint is optional with TFS and not covered as part of the proof-of-concept.<br>• Microsoft Azure Architectures for SharePoint 2013 [60]<br>• Building a disaster recovery environment for SharePoint in Microsoft Azure [61]<br>• SharePoint Server 2013 Disaster Recovery in Microsoft Azure [62]<br>• Internet Sites in Microsoft Azure using SharePoint Server 2013 [63]<br>• Microsoft Azure Active Directory with SharePoint 2013 [64]<br>• SQL Server in Microsoft Azure Virtual Machines [65]<br>• High Availability and Disaster Recovery for SQL Server in Microsoft Azure VMs [66]<br>• Configuring AlwaysOn Availability Groups [67] |
| 8<br>Review Performance<br>☐ - Done | Review and consider performance learnings to improve the reliability and performance.<br>• Review section **Performance Learnings**, page 21.<br>• Review Azure Forums and contact an Azure, TFS and SQL subject matter expert. |
| 9<br>Validate environment<br>☐ - Done | Before proceeding, please validate your target Azure and dependent environments.<br>• Verify that your **network is reliable** in terns of availability, security and quality.<br>• Verify that no **Azure maintenance** is scheduled that could potentially impact your deployment.<br>• Record all "known" TFS and connectivity issues, so that "new" issues can be cross checked. |
| 10<br>Validate<br>☐ - Done | Contact an Azure subject matter expert from Microsoft or the Azure Community [68] and validate the plans before committing yourself. Ensure that the planned environment is technically and financially viable, and can be maintained. |

**Table 5 – Planning Checklist**

[59] http://blogs.msdn.com/b/dfurman/archive/2014/04/27/using-storage-spaces-on-an-azure-vm-cluster-for-sql-server-storage.aspx

[60] http://technet.microsoft.com/en-us/library/dn635309(v=office.15).aspx

[61] http://blogs.technet.com/b/tothesharepoint/archive/2014/02/28/a-few-things-we-learned-by-building-a-disaster-recovery-environment-for-sharepoint-in-windows-azure.aspx

[62] http://technet.microsoft.com/en-us/library/dn635313(v=office.15).aspx

63 http://technet.microsoft.com/en-us/library/dn635307(v=office.15).aspx

[64] http://technet.microsoft.com/en-us/library/dn635311(v=office.15).aspx

[65] http://msdn.microsoft.com/en-us/library/windowsazure/jj823132.aspx

[66] http://msdn.microsoft.com/en-us/library/windowsazure/jj870962.aspx

[67] http://msdn.microsoft.com/en-us/library/windowsazure/dn275964.aspx

[68] http://azure.microsoft.com/en-us/support/forums/

# Machine Planning

The machine-planning checklist consolidates the machine details, such as domain affinity, name, size, and role in one place for easy reference when setting your environment.

## Machine cheat sheet

| Domain Location | Name | Core | IP (int/public) | Affinity | Role |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

**Table 6 – Machine cheat sheet**

## Machine cheat sheet – POC sample

| Domain Location | Name | Cores | IP (int/public) | Affinity | Role |
|---|---|---|---|---|---|
| TFSAZPOC Azure | AZ-UE-AD-O1 | 1 | 10.0.0.4<br>137.116.32.110 | TFS-Affinity-Group | Domain controller |
| TFSAZPOC Azure | AZ-UE-AT-O1 | 2 | 10.0.0.6<br>137.116.32.177 | TFS-Affinity-Group | Application Tier |
| TFSAZPOC Azure | AZ-UE-DT-O1 | 4 | 10.0.0.5<br>137.116.32.70 | TFS-Affinity-Group | Data Tier |
| TFSAZPOC Azure | AZ-UE-BS-O1 | 2 | 10.0.0.7<br>137.116.32.182 | TFS-Affinity-Group | Build Server |
| TFSHOLPOC on prem | OP_HOL_AD_01 | | 192.168.0.1 | On premises | Domain controller |
| TFSHOLPOC on prem | OP_HOL_GW_01 | | 192.168.0.254<br>205.196.30.66 | On premises | Gateway |
| TFSHOLPOC on prem | OP_HOL_FS_01 | | 192.168.0.10 | On premises | File server |

**Table 7 – Machine cheat sheet – POC sample**

# Credentials Planning

The credentials planning checklist consolidates the user accounts and passwords in one place, for easy reference while setting up your environment.

## Cheat sheet

| Domain | User | Password | Role |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

**Table 8 – Credentials cheat sheet**

## POC example

| Domain | User | Password | Role |
|---|---|---|---|
| . | TFSAzurePocUser | TfsPoc@1 | Local admin for POC server |
| TFSAZPOC | TFSAzurePocUser | TfsPoc@1 | Enterprise admin for POC |
| TFSHOLPOC | Chris Margraff | ############## | Domain administrator |
| TFSHOLPOC | Hassan Fadili | ############## | Domain administrator |
| TFSHOLPOC | Michael C. Bazarewsky | ############## | Domain administrator |
| TFSHOLPOC | Willy-Peter Schaub | ############## | Domain administrator |
| TFSHOLPOC | Tony Feissle | ############## | Domain administrator |
| TFSHOLPOC | TFSUser | TfsPoc@1 | This account simulates a generic TFS user that will have access to the environment. Since the account has no other privileges, you should test your configuration by logging in without administrator privileges. |
| TFSHOLPOC | TFSAdmin | TfsPoc@1 | This account has administrative rights over the entire TFS deployment. Use it to set up projects and workspaces. |
| TFSHOLPOC | TFSBuild | TfsPoc@1 | Service account for Team Foundation Build |
| TFSHOLPOC | TFSService | TfsPoc@1 | Service account for Team Foundation Server |
| TFSHOLPOC | TFSSystem | TfsPoc@1 | The TFS timer service uses this account to start scheduled processes (such as builds). |

**Table 9 – Credentials cheat sheet – POC sample**

# Deployment

The deployment checklists guide you through the deployment process of the TFS Azure IaaS environment, based on our proof of concept (POC).

| Step | Instructions |
|---|---|
| * Planning ⬚ - Done | • Planning checklist, page 23,<br>• Machine checklist, page 25, complete.<br>• Credentials checklist, page 26, complete. |
| 1 Explore on-premises Environment ⬚ - Done | • Explore the **on-premises** environment as a proof of concept.<br>• See page 34 for a journal of the POC **On-Premises Environment**. |
| 2 Azure Subscription Environment ⬚ - Done | • Create the **Azure Subscription** environment.<br>• See page 35 for a journal of the POC **Azure Subscription Environment**. |
| 3 Azure Network ⬚ - Done | • Summarize all **machine** names, IP addresses, location, and role for easy reference. See page 25 for a template and POC example.<br>• Summarize all **credentials** for easy reference. See page 26 for a template and POC example. |
| 4 Prepare all pre-requisite software ⬚ - Done | • Ensure you have the media or download URLs for the required software:<br>  • Team Foundation Server<br>  • Visual Studio<br>  • SQL Server |
| 5 Azure Network ⬚ - Done | • Create the **Azure Network** environment.<br>• See page 37 for a journal of the POC **Azure Network** environment. |
| 5 Azure Storage ⬚ - Done | • Create the **Azure Storage** environment.<br>• See page 47 for a journal of the POC **Azure Storage**. |
| 7 Servers in Azure ⬚ - Done | • Create the **servers** needed in the Azure environment.<br>• See page 48 for a journal on the POC **Domain Controller**.<br>• See page 59 for a journal on the POC **Data Tier (DT) Server**.<br>• See page 65 for a journal on the POC **Application Tier (AT) Server**.<br>• See page 70 for a journal on the POC **Build Server (BS) Walkthrough**. |
| 7 Functional Test ⬚ - Done | • Perform a comprehensive test plan to validate your environments and the TFS services. See page 73 for an example **Test Plan**. |
| 8 Performance Test ⬚ - Done | • Perform a comprehensive performance test plan to validate the environment for your current and projected service loads. |

**Table 10 – Deployment Checklist**

# Companion Proof-Of-Concept (POC)

GOAL

We created our own TFS on Azure IaaS environment to evaluate, research, and validate our guidance assumptions and planning decisions. This section introduces the ALM Rangers planning of the proof-of-concept (POC) research environment, elaborates on a few pitfalls, and describes the final POC environment.

## Current hypothetical environment


NOTE

We are simulating an Azure environment, connected with an on-premises domain. See Deploying TFS completely in the Cloud [69] if you are looking for an Azure only environment.

**Trey Research** is an application development company working on specialized out-of-band solutions for numerous customers and the ALM community. Most engineers are working from home, on-premises and travelling to customers in North America and Europe, using TFS as their ALM solution.

The company currently has **100** employees, and is not planning to grow beyond **250** employees interacting with TFS.

Their current TFS environment consists of an active directory, a dual-server TFS deployment in North America, and a TFS Proxy server in Europe, with a centralized file server used to drop all builds.

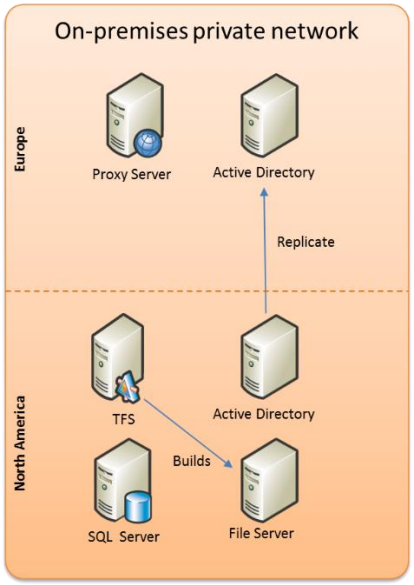



**Figure 3 – Trey Research on-premises environment**

[69] http://blogs.msdn.com/b/chrismargraff/archive/2014/05/20/deploying-tfs-in-the-cloud.aspx

# Moving to Azure

Microsoft Azure has different capabilities, so Trey Research cannot just move the on-premises TFS instance "as-is" to the cloud. As part of the proof-of-concept planning, we considered three scenarios.

## Scenario 1: Move everything to separate Azure regions

Moving the environment to three **separate** Azure regions to replicate the on-premises, European, and US environments is feasible, but not practical.

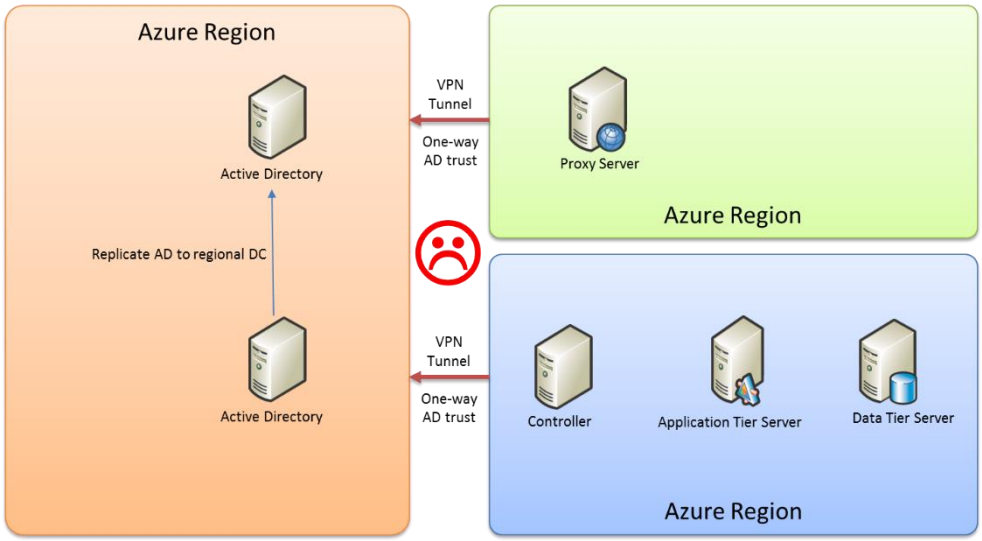



**Figure 4 – Move everything to Azure IaaS**

In theory this scenario is possible if we were to consider other service providers for the Azure US Region environment on the left. However, using Azure support of connectivity across regions is limited "by design" and we will not be able to VPN cross-data-centers today. In the future, using Multiprotocol Label Switching (MPLS) connections of certain Telecommunication providers, you may be able to do this, but it is not available outside the US today.

EXPLANATION

Eric, our Azure subject matter expert, explains:

*All of the network traffic through the datacenter, from a public endpoint on the load balancer to a physical box, hosting virtualized networks with virtualized servers, etc. is all NAT'ed. There is a plethora of software-based routing and network layer translations that happen.*

*For that reason, an IP/port in one service in one data center (regions) would never be recognizable, and hence would be blocked via rules by the Azure Fabric Controllers controlling all the traffic translation and addressing. Therefore, even though you could setup your own server-to-server VPN, the traffic itself simply would not make it across the stack. This translation actually happens at several layers in the network stack, so you are talking Layer 0 -7 all-inclusive in some cases, or in others, just the higher layers at the application level. So, in the end, each data center's containing services are pretty well isolated from each other's, which is what makes some of the features of Azure like geo-replication and redundancy cross-DC so complex and difficult to architect and provide.*

## Scenario 2: Move everything to one Azure region

Moving the environment to **one** Azure is feasible, but does not replicate the replicate the on-premises European and US environments. While practical for a regional team, this scenario is not practical for geographically distributed teams.

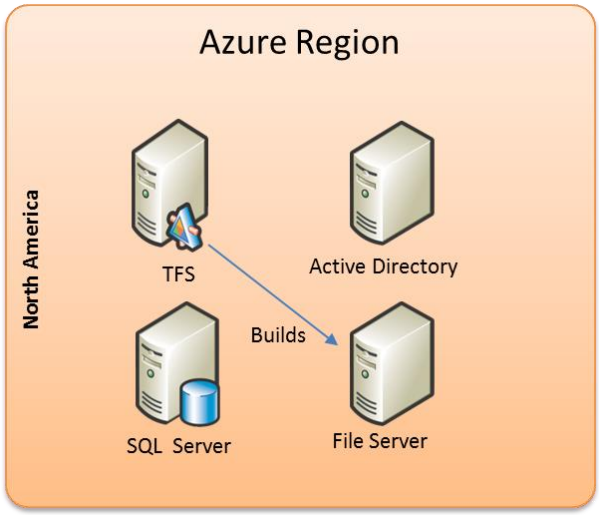



**Figure 5 - Move everything to one Azure region**

## Scenario 3: Keep AD on-premises and move TFS to Azure IaaS

The scenario we opted for was to simulate the on-premises domain, responsible for all user authentication, and two regions, one in North America and one in Europe as part of one Azure Subscription. Connectivity is created using Site-To-Site VPN from on-premises to the region in North America and Europe, whereby the latter is not covered in this guide. Refer to the **TFS on Azure IaaS v1.4 Supplement - Improve performance for remote teams with TFS Proxy Server** guide, included in the guidance downloads, for more information.

REVIEW

More from Eric:

*Theoretically, we could tunnel from the Europe data center, to an on-premises environment, then back to the Azure DC in the US, but for purposes of Active Directory, replication, etc. it just is not practical, and does not work. We have had discussions with networking teams about RADIUS servers and other approaches, but it simply will not work out.*

The proposed POC environment is therefore, (as shown with an on-premises data center hosting the Active Directory and File Server) an Azure Europe region hosting the TFS Proxy server and an Azure US region hosting the TFS environment.

One-Way trust from the Azure network to the on-premises network is accomplished through a VPN tunnel using Routing and Remote Access Service (RRAS) on Windows Server 2012. You could choose other Windows Server versions, but we decided to consistently use Windows Server 2012 across the board.

# Plan the TFS environment

Using the TFS Planning and DR Avoidance Guide [70], as outlined in **Planning Environment**, page 18, and **Sizing and mapping VMs** page 18, we note that ❶ we could use a single server for both the application and data tier. Instead, we opt to have the tiers separated to cater for future growth using the ❷ scale-up scenario. Review the

[70] http://aka.ms/treasure5

TFS Planning and DR Avoidance Guide [71] for an explanation of the three architectures and how the minimum recommendations are calculated.

**Visual Studio ALM Rangers - On-Premises Capacity Planning Quick Reference Poster Companion Workbook**

*Version 2012.02.24*

**NOTE:** *This is a simplified model for educational purposes and does not replace expert consulting. Real-world requirements and infrastructure environments can be much more complex.*

| NUMBER OF ACTIVE TEAM FOUNDATION SERVER USERS | |
|---|---|
| Maximum expected users | 250 |
| Current users | 100 |

| Real-World (Beef) Factor | 0% |
|---|---|

❶ ❷

| INFRASTRUCTURE ARCHITECTURE | Single Server ATDT | Scale Up AT + DT | Scale Out AT + DT |
|---|---|---|---|
| Feasible recommendation for current? | Yes | Yes | No |
| Feasible recommendation for future? | Yes | Yes | No |
| Preferred Architecture? | Yes | Yes | No |
| Server Configuration Type | Low-End - Low-End | High-End | Scale Unit |
| Maximum Current Active Collections (per SQL Server instance) | 5 | 75 | |
| Maximum Future Active Collections (per SQL Server instance) | 5 | 75 | |
| Estimated number of Application Tier (AT) servers | 1 | 1 | |
| Estimated requests per second (rps) in total | 18 - 44 | 23 - 56 | |
| Estimated requests per second (rps) on AT server | | | |

Visual Studio ALM Rangers

**Figure 6 – Using the capacity-planning workbook: server scenarios**

Switching to the hardware configuration section we select the ❷ scale up scenario and ❸ virtual as we will be hosting on Azure IaaS virtual machines.

**Visual Studio ALM Rangers - On-Premises Capacity Planning Quick Reference Poster Companion Workbook**

Version 2012.02.24 ❸

**NOTE:** *This is a simplified model for educational purposes and does not replace expert consulting. Real world requirements and infrastructure environments can be much more complex.*

| | MINIMUM CONFIGURATION FOR CURRENT REQUIREMENTS | Single Server | | Scale Up | | | |
|---|---|---|---|---|---|---|---|
| | | Physical | Virtual | Physical | | Virtual | |
| | | ATDT | ATDT | AT | DT | AT | DT |
| BASE RECOMMENDATIONS | CORE | 1 | 0 | 2 | 4 | 2 | 5 |
| | RAM (GB) | 2 | 2 | 4 | 8 | 5 | 10 |
| | DISK (GB) | 125 | 150 | 500 | 2000 | 600 | 2400 |
| | **MINIMUM RECOMMENDED CONFIGURATION FOR FUTURE GROWTH** | **Single Server** | | **Scale Up** | | | |
| | | Physical | Virtual | Physical | | Virtual | |
| | | ATDT | ATDT | AT | DT | AT | DT |
| | CORE | 1 | 1 | 2 | 4 | 2 | 5 |
| | RAM | 2 | 2 | 4 | 8 | 5 | 10 |
| | DISK (GB) | 125 | 150 | 500 | 2000 | 600 | 2400 |

**Figure 7 – Using the capacity-planning workbook: server sizing**

[71] http://aka.ms/treasure5

# Map to Azure IaaS environment

As discussed in **Sizing and mapping VMs**, page 18, there is no perfect mapping between Azure VMs and physical machines. We opted closet match with the lowest Azure subscription costs for the POC.

| Server | Planning Guide Recommendation | Azure VM Configuration |
|---|---|---|
| Application Tier | 2 Core 5GB RAM 500GB Disk | Medium 2 Core 3.5GB |
| Data Tier | 5 Core 10GB RAM 2TB Disk | Large 4 Core 7GB |
| Build Server | Not calculated by planning guide | Medium 2 Core 3.5GB |
| Proxy Server | Not calculated by planning guide | Medium 2 Core 3.5GB |

**Table 11 - Server Requirements Planning**

# POC naming conventions

## Machines

The naming conventions used in our proof-of-concept (POC) are summarized as follows:
**LOCATION_REGION_ROLE_INSTANCE** for on on-premises and
**LOCATION-REGION-ROLE-INSTANCE** for Azure environments, as underscore were not allowed in Azure names.

| LOCATION | | REGION | | ROLE | | INSTANCE |
|---|---|---|---|---|---|---|
| On-premises | OP | On-Premises | HOL | Active Directory | AD | We planned to use two digits to identify the instance of a specific role, i.e. 01, 02, … |
| Azure IaaS | AZ | Azure US East | UE | Domain Controller | DC | |
| | | Azure North Europe | NEU | Application Tier | AT | |
| | | | | Data Tier | DT | During actual setups we also used an O instead of a 0 for the first digit to avoid name lashes with re-installs. |
| | | | | Build Server | BS | |
| | | | | Proxy Server | PX | |
| | | | | Gateway | GW | |

**Table 12 – POC naming conventions**

Example: OP_HOL_AD_01 is the name for the first Active Directory server in the on-premises domain.

## Domain Names

For domain names we continued with simplicity using **TFSHOLPOC** for on-premises domain name and **TFSAZPOC** for Azure domain name

# POC Architecture "at a glance"

The POC architecture we are implementing looks as follows:

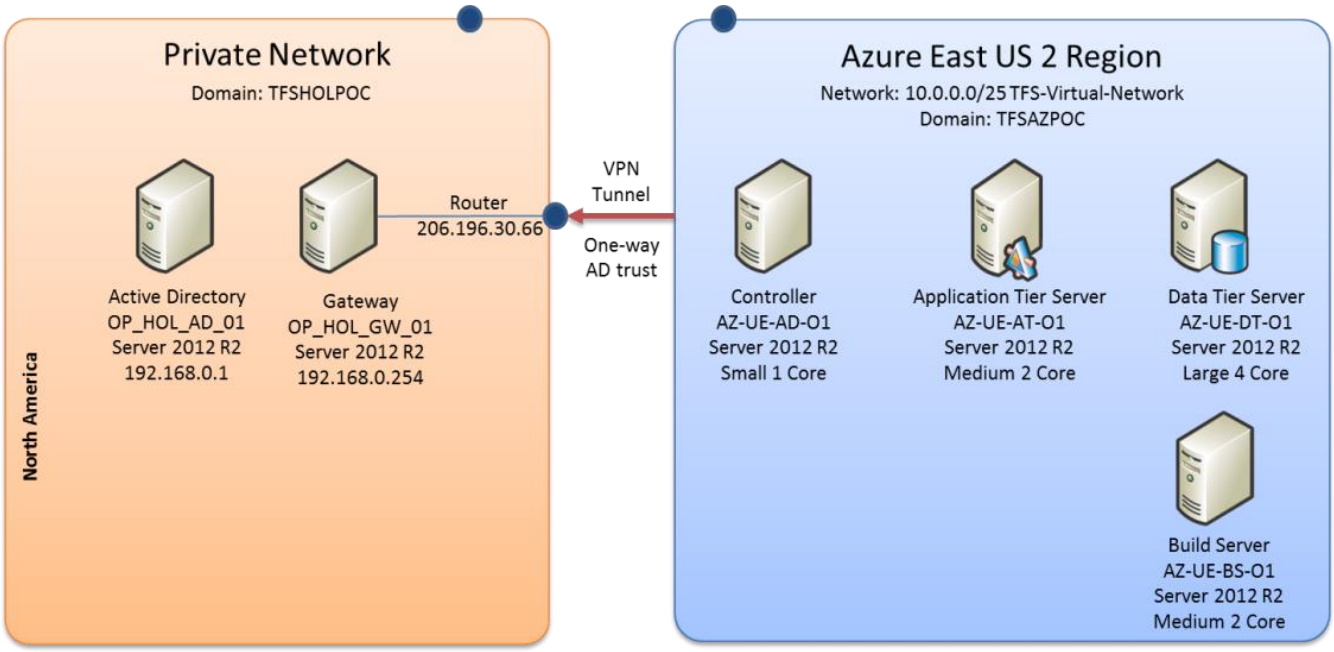



**Figure 8 - POC Architecture "at a glance"**

We base the resultant proof-of-concept (POC), as shown above, on a real-world scenario. You can replicate it for a research or development environment and extended to support geographically distributed development teams.

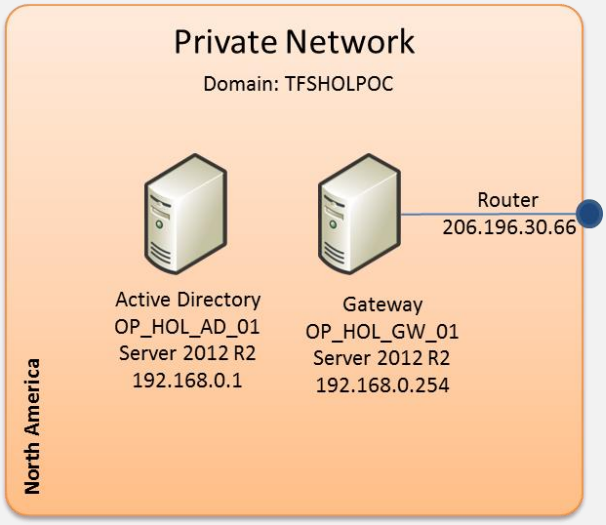
# Deployment of POC

In the following and supplementary walkthroughs, we will take you through all the setup and configuration steps. The walkthroughs were created while setting up the proof-of-concept and serve as handy journals you can reference when implementing your TFS on Azure IaaS environments.

WARNING

Microsoft Azure, Windows, and Team Foundation Server environments are rapidly evolving and the screenshots in these walkthroughs might differ from what you see in your environment!

## On-Premises Environment

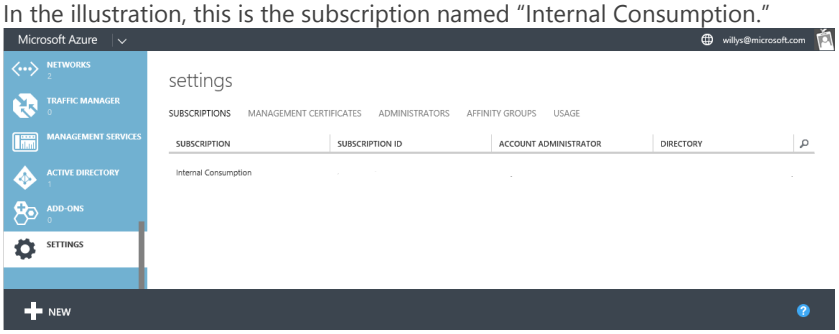
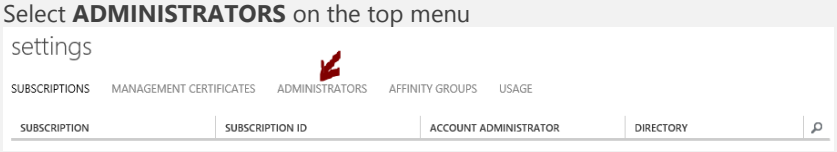
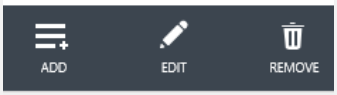
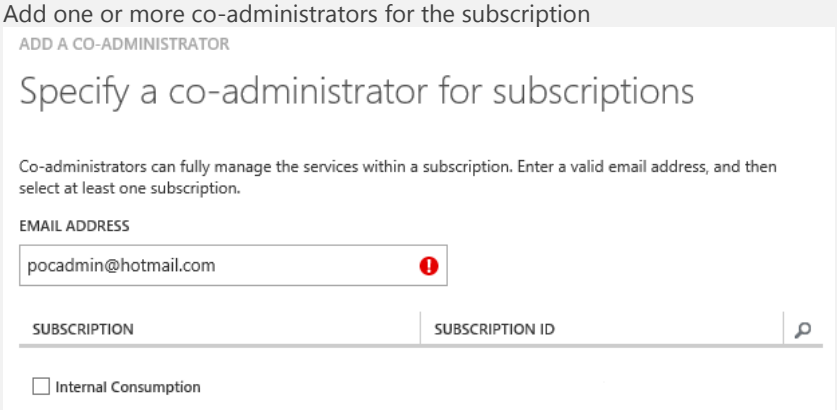
| Step | Instructions |
|---|---|
| 1<br>Select environment<br>☐ - Done | • We do **not** recommend the use of production on-premises environment for a proof-of-concept.<br>• We worked with holSystems [72] to implement an isolated, real world, on-premises environment, located outside of the Azure cloud.<br>holSystems<br>• The use of virtualized environments allows us to snapshot and rollback the on-premises environment as needed. |
| 2<br>Understand environment<br>☐ - Done | • Explore the on-premises environment to verify administrative access and validate the infrastructure.<br>  ○ OP_HOL_GW_01: RRAS gateway server. External IP 206.196.30.66, internal IP 192.168.0.254<br>  ○ OP_HOL_FS_01: File server, 192.168.0.10<br>  ○ OP_HOL_AD_01: Domain controller, 192.168.0.1<br>  ○ RDP is available on the OP_HOL_GW_01<br>  ○ Hop from OP_HOL_GW_01 to the FS and DC<br>• Visualize the private on-premises environment and share with all POC stakeholders.<br>  |

[72] http://www.holsystems.com/

| Step | Instructions |
|---|---|
| 3<br>Define ownership<br>☐ - Done | • Assign ownership and access to your team.<br>  ○ We constrain access to on-premises environment to the owner of the Azure POC environment and one subject matter expert.<br>• Notify the team who and how to contact with queries or requests for the in-premises environment. |
| 9<br>Snapshot<br>☐ - Done | • Snapshot the on-premises environment to create an initial checkpoint. |

**Table 13 – Prepare on-premises POC environment**

# Azure Subscription Environment

| Step | Instructions |
|---|---|
| 1<br>Validate Azure subscription<br>☐ - Done | • Logon to https://manage.windowsazure.com and either create a subscription or select the subscription created and assigned to you for the proof-of-concept (POC).<br>• In the illustration, this is the subscription named "Internal Consumption."<br>  <br>• Select **Settings** in the navigation bar to view your subscription settings. |
| 2<br>Create Admins<br>☐ - Done | • Select **ADMINISTRATORS** on the top menu<br>  <br>• Select **ADD** on the bottom menu<br>  <br>• Add one or more co-administrators for the subscription<br>  |

| Step | Instructions |
| --- | --- |
| 3<br>Notify Admins<br>⬚ - Done | • Notify the co-administrators and verify that they can access the subscription. |

**Table 14 –Azure subscription environment**

# Azure Network

## Whiteboard Diagrams

In this walkthrough, we will create the following subject matter expert's whiteboard diagram in the Azure environment.

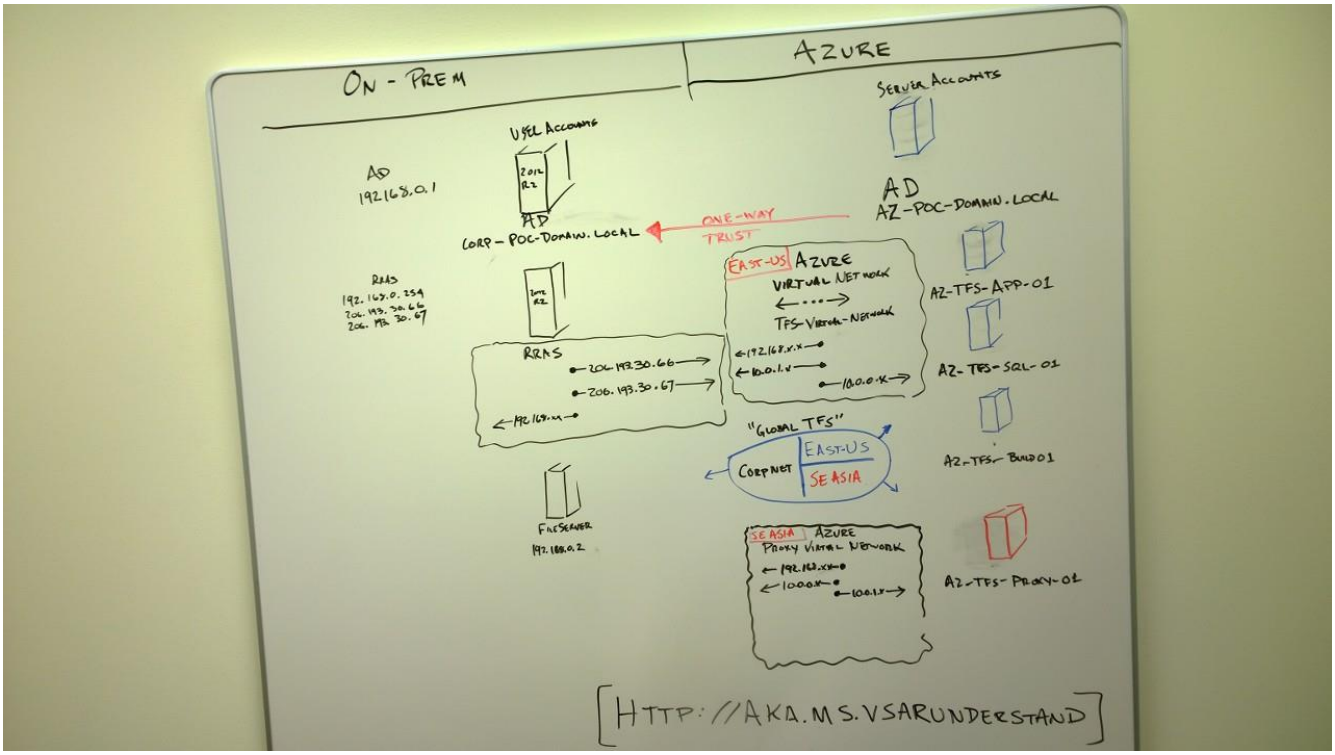



**Figure 9 - Photo 1 of SME network diagram on whiteboard**

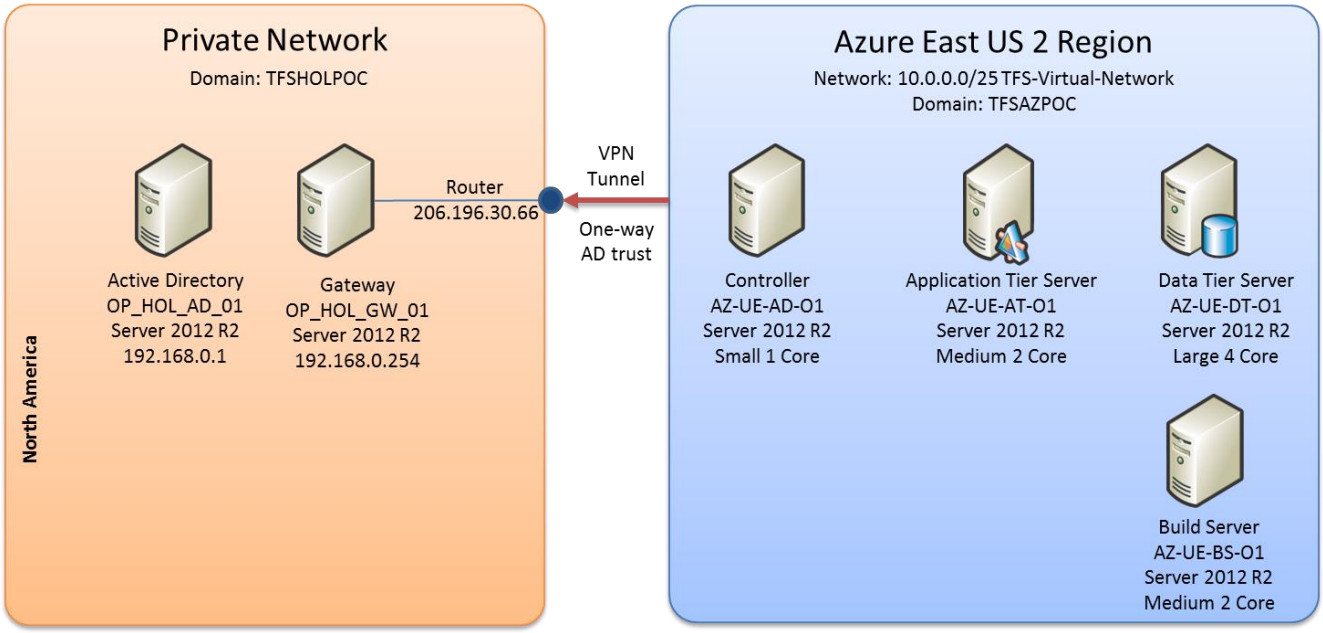
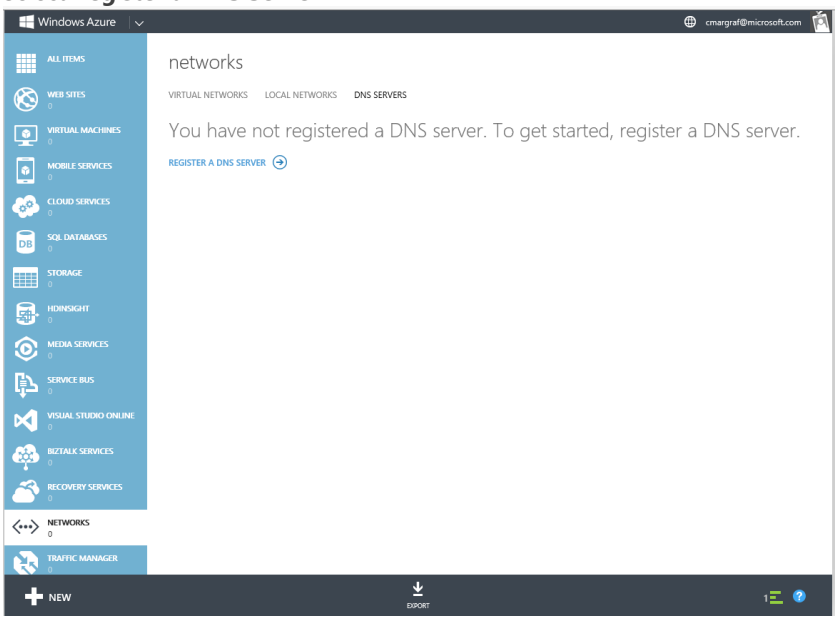
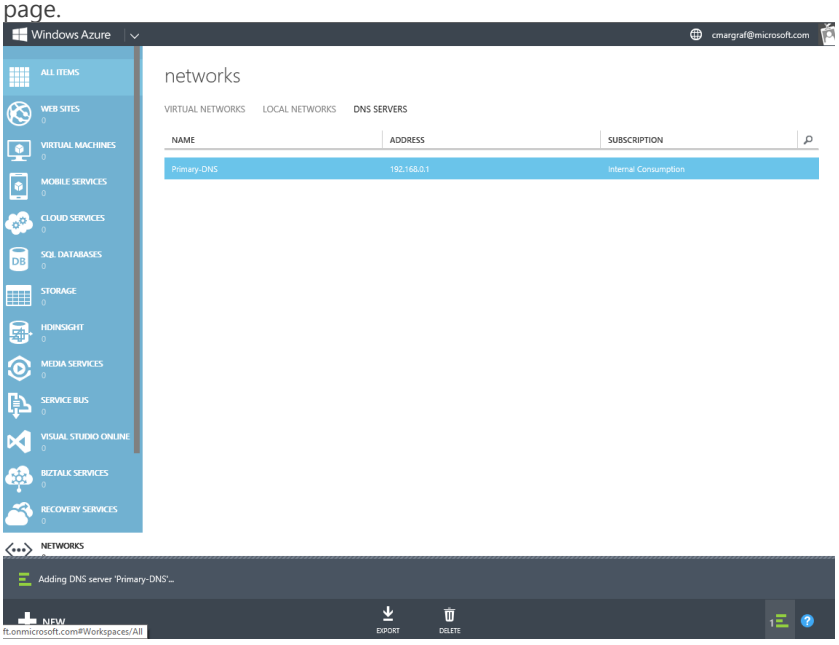
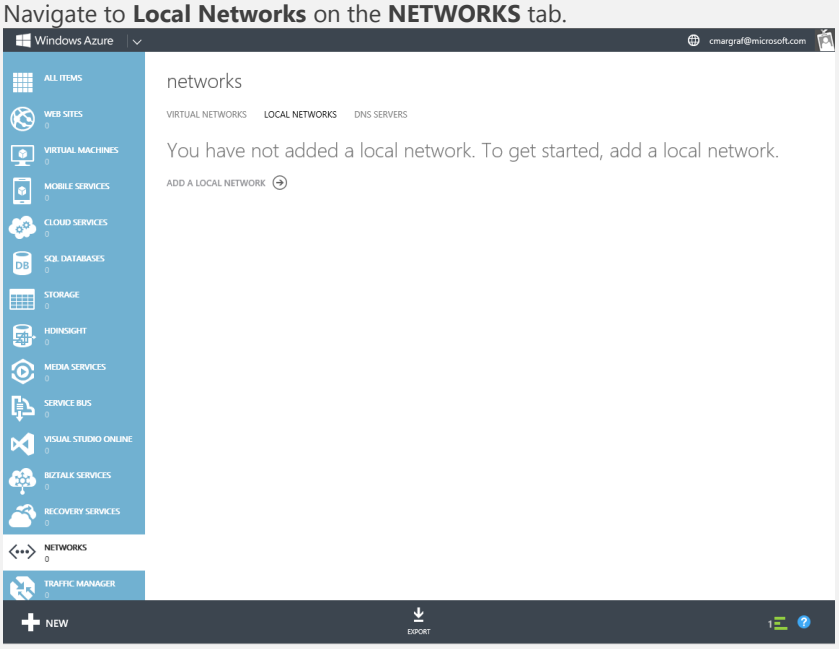



**Figure 10 - High-altitude network diagram**

# Walkthrough

| Step | Instructions |
|---|---|
| 1 Register DNS Server for POC computer name resolution ☐ - Done | • Navigate to DNS Servers on the **Networks** Tab.<br>• Select **Register a DNS Server**<br>  <br>• Enter a **name** for this registration: *Primary-DNS*<br>• Enter the **DNS Server IP**: *192.168.0.1***.** This is the IP address of your Active Directory domain controller, which has DNS enabled.<br>NOTE: You will not be able to resolve this until the virtual network is completed and connected.<br>• Click **Register**.<br>• The Registration will take a few seconds to complete. A status bar will appear at the bottom of the page.<br>  |

| Step | Instructions |
|---|---|
| 2 Register Local Network ⬚ - Done | • Navigate to **Local Networks** on the **NETWORKS** tab. |



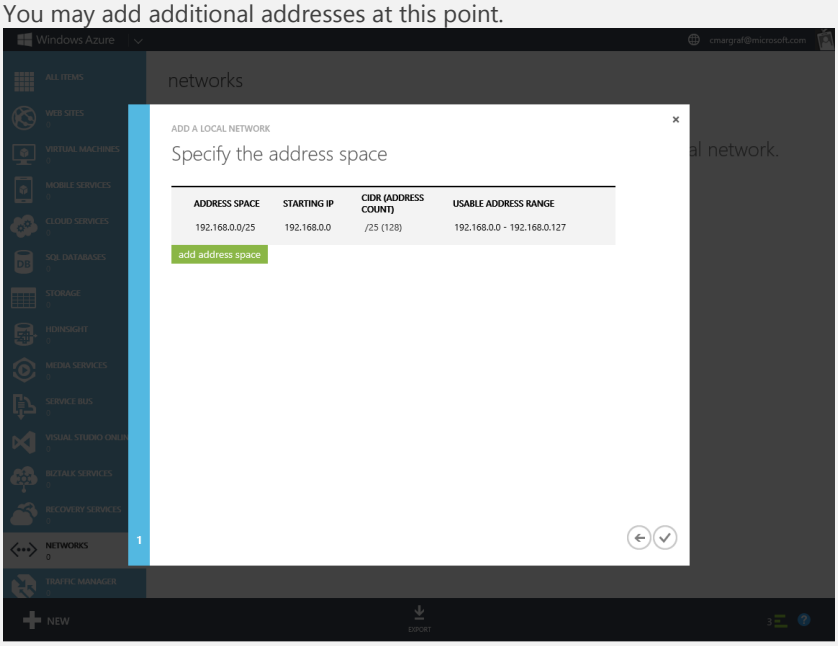

- Click **Add a Local Network**.
- **Name** the local network *Gateway-GW-01*.
  This name will refer to your on-premises gateway providing the VPN tunnel.
- Enter the public **IP address** assigned to the gateway. This public address will vary in every deployment.
- Click **Next**.
- Add the address range **192.168.0.0/25**

NOTE

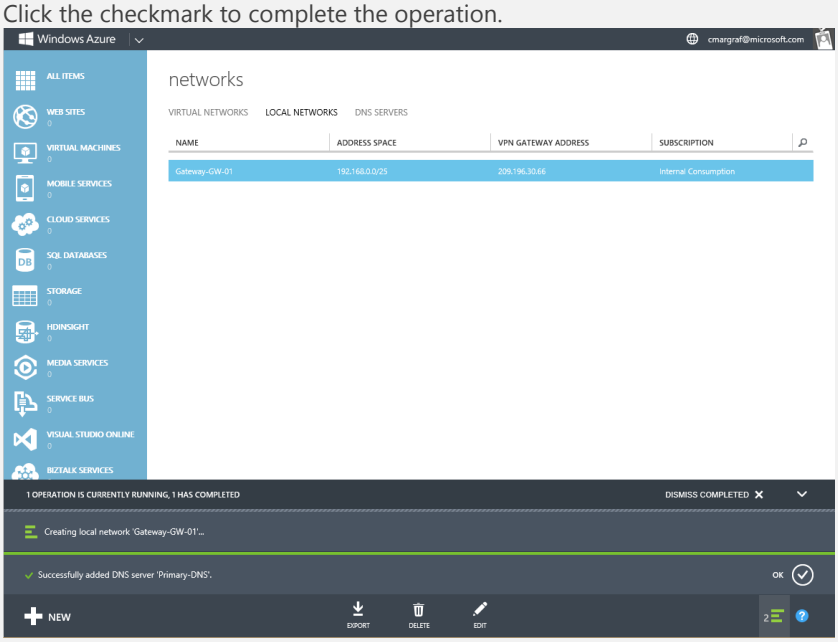
The value you enter here will be the range of addresses on the existing **client-side local network** that can connect to Azure. The Azure gateway will route traffic destined for this address range through the VPN tunnel. Note that we are using non-routable addressing for the on-premises environment. The VPN tunnel will handle routing of this traffic.

| Step | Instructions |
|---|---|

- You may add additional addresses at this point.



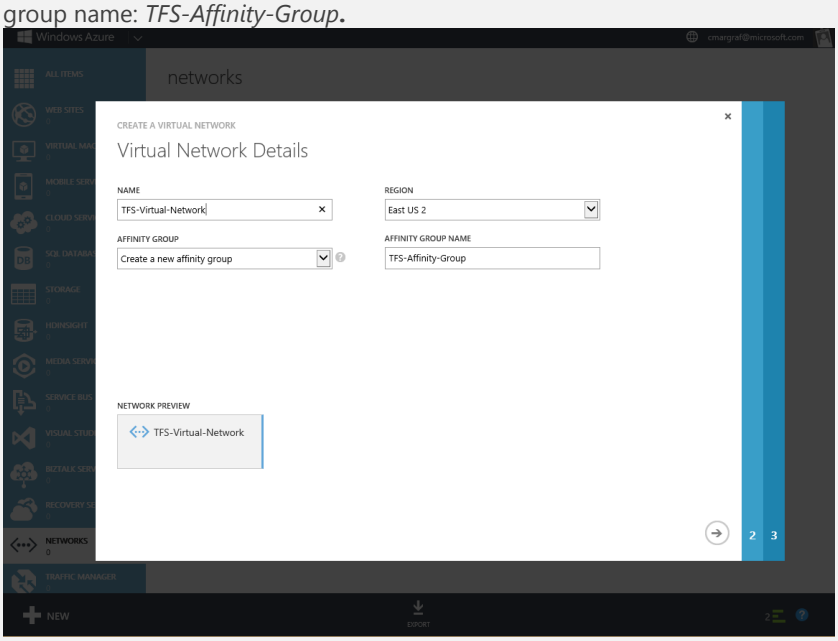

- Click the checkmark to complete the operation.



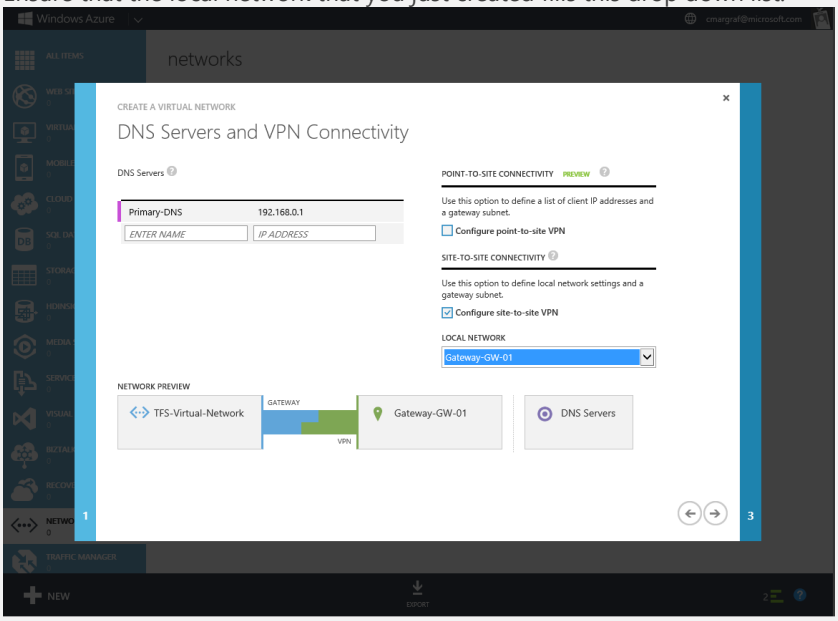

- You should see a notification message of successful completion.
- Click **Virtual Networks** on the **Networks** tab.
- Click **Create a Virtual Network**.
- Name the network **TFS-Virtual-Network**.
- Select the region which this network will be deployed in: *East US 2*

| Step | Instructions |
|---|---|

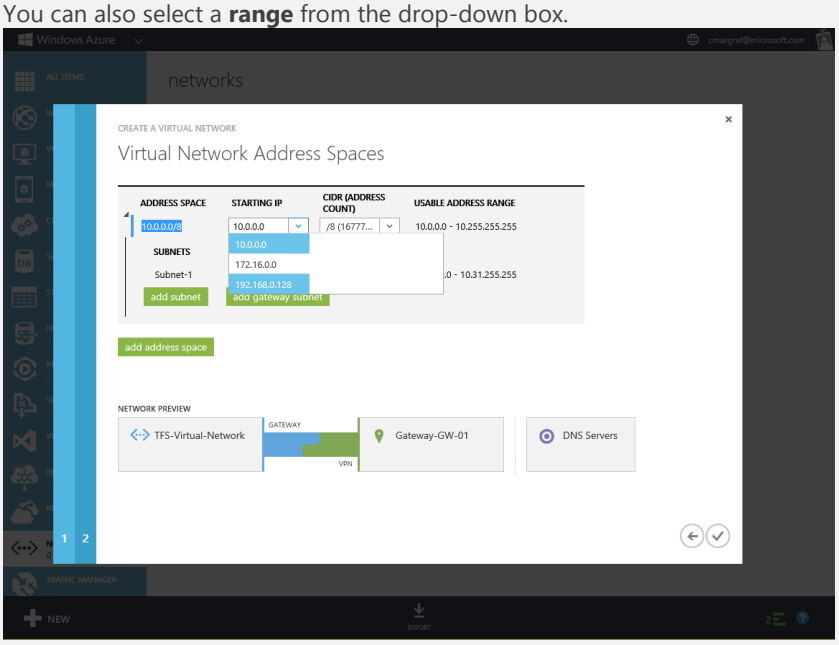
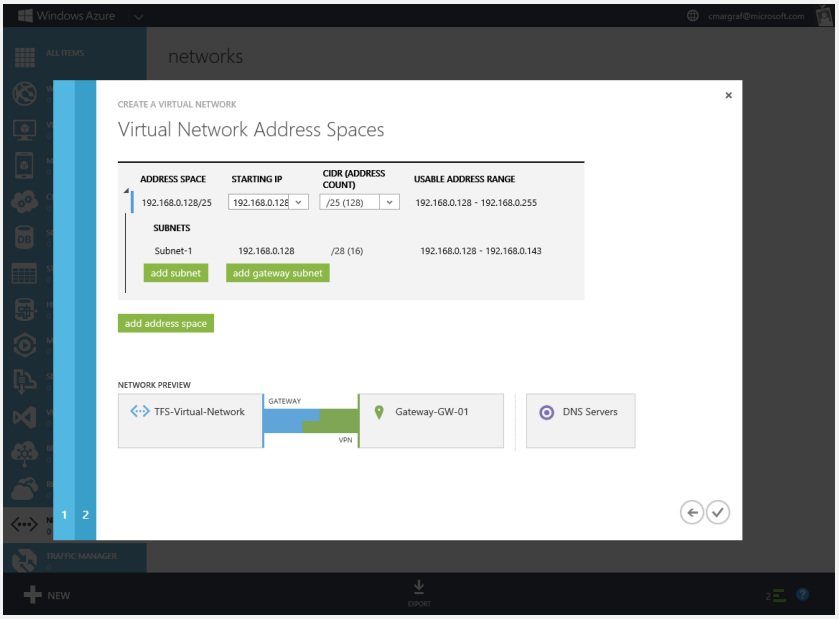
- You will create a new affinity group. Leave the drop-down list at its default, and enter an affinity group name: *TFS-Affinity-Group*.

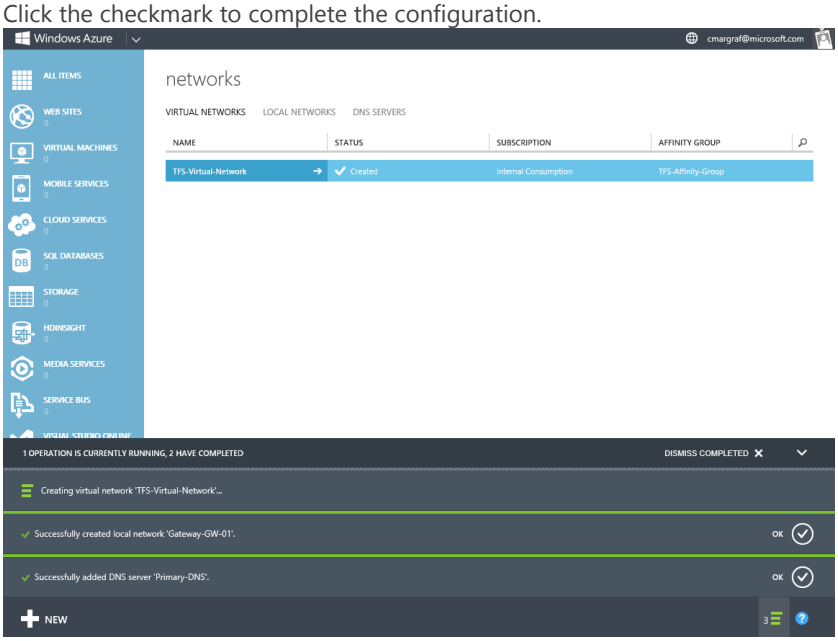
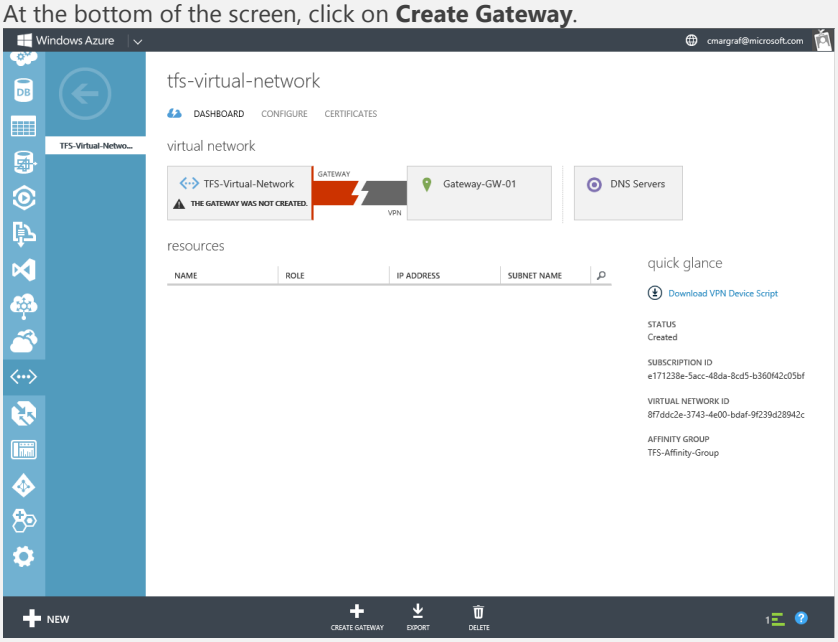


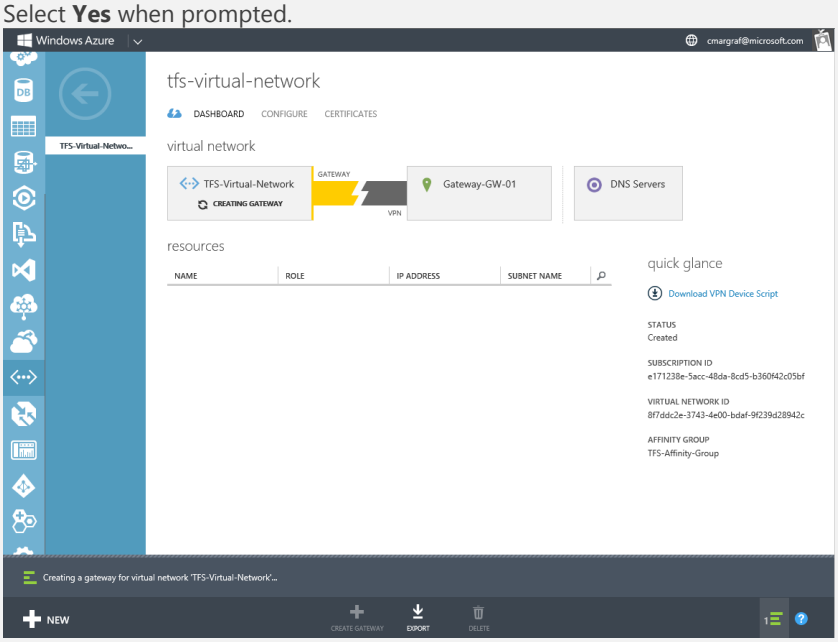
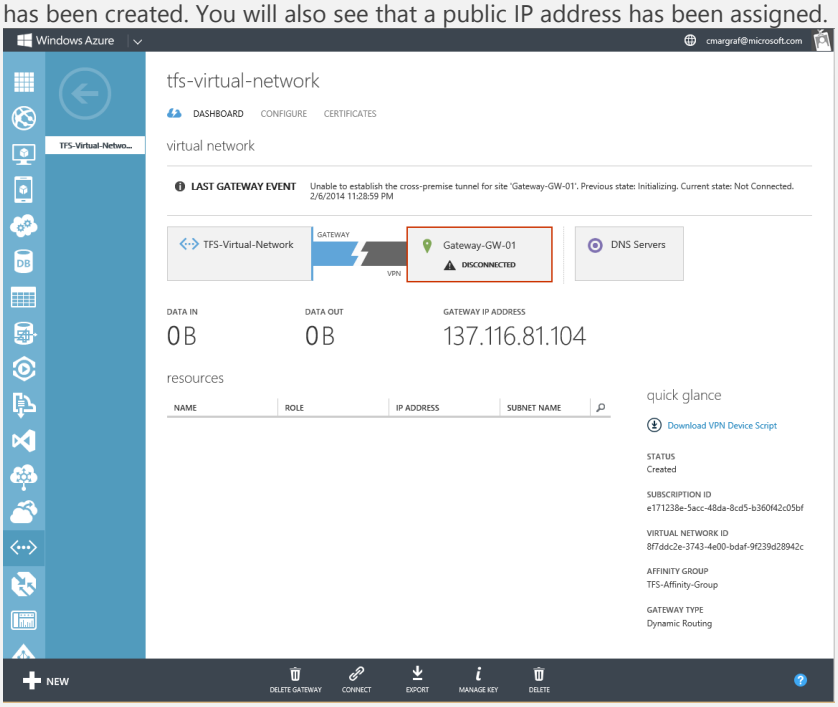

- Click **Next**.
- Select the **name** of the DNS Server that you just registered from the drop-down list.
- Check the box to configure site-to-site VPN.
- Leave the point-to-site checkbox unchecked.
- The local network will appear in a drop down list below the site-to-site checkbox.
- Ensure that the local network that you just created fills this drop down list.

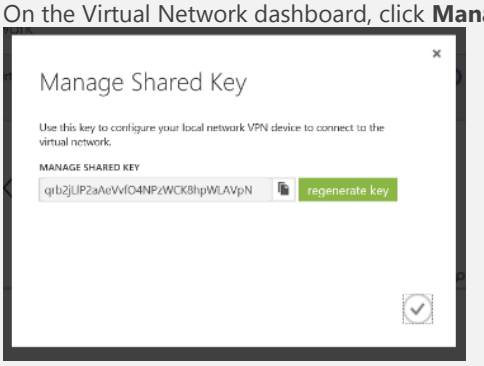



- Click **Next**.
- Then next page is the virtual network address space. This is the address space assigned to devices within the current Azure subscription.
- Select the drop-down titled ***Starting IP***.
- We used the default IP address range provided *10.0.0.0/8*.

| Step | Instructions |
|---|---|
| | NOTE We consulted the Hybrid Cloud Center of Excellence (CoE) for a recommendation on which type of network segment to use. They recommended we use non-contiguous network segments when possible. This allows expansion of IP address ranges on both sides of the connection without a complete rebuild. In our case, this also lets us accept all default values on the Virtual Network address space configuration page.<br><br>• You can also select a **range** from the drop-down box.<br>  <br>• After selecting, note the default **10.0.0.0** address space, which reflects your selection.<br>• Set the **address count**. We used the default. You should select a minimum of eight IP addresses for this POC.<br>  |
| 3 Configure Subnets<br>☐ - Done | • In the subnet1 Address Count box, select a segment of the address space you just added*: /27 (32)*, whereby we accepted the default first subnet here.<br>• **Name** Segment-1: *TFS-Farm-Seg*<br>• Click **Add Subnet** to create another subnet. |

| Step | Instructions |
|---|---|
| | • Leaving the Starting IP on the new row at its default value, enter another address count: */29(8)*. We accepted the default values presented for the second subnet.<br>• **Name** the subnet: *TFS-Proxy1-Seg*<br>• Click **Add Gateway Segment**.<br>• Set the Gate segment's CIDR: */29(8)*<br>• Click the checkmark to complete the configuration.<br>  |
| 4<br>Setup the Azure side gateway<br>⬚ - Done | • Click **Virtual Network** on the **Network** tab.<br>• Click on the Virtual Network you created.<br>• Click on **Dashboard**.<br>• At the bottom of the screen, click on **Create Gateway**.<br>  <br>• You will need to choose between static and Dynamic Routing.<br>• Select **Dynamic Routing**. |

| Step | Instructions |
|---|---|
| | • Select **Yes** when prompted.<br>  <br>• It will take some time to create the Gateway. It took 15 minutes in our lab. You can navigate away from this page and return later. The "Last Gateway Event" notification will indicate when the gateway has been created. You will also see that a public IP address has been assigned.<br>  |
| 5<br>Configure the on-premises RRAS gateway<br>☐ - Done | • Once you have created the gateway, you can export the VPN device script to reflect this configuration on the VPN device.<br>• Click **Download VPN Script**.<br>• Select the device you are using<br>• Select **Export**.<br>• You will receive a device-specific configuration file.<br>• In the case of Windows 2012 RRAS, this will be a PowerShell script. See the **Export_VPN_Script.ps1** script included in the guidance downloads from our POC for an example. |

| Step | Instructions |
|---|---|
| | • Your on-premises gateway should **NOT** have RRAS installed at this time. The script you are about to run will install RRAS for you. If RRAS is installed, disable and uninstall it at this time if you want to use the PowerShell script.<br>• Open the PowerShell IDE running with Administrator privileges.<br>• Run the following command: set-ExecutionPolicy Unrestricted<br>• Paste the script you downloaded into the PowerShell IDE and execute it.<br>• If RRAS is not installed, the commands at the end of the installation script fail execution. This is because you must reboot your gateway server before continuing.<br>• Reboot your Gateway Server. After logging back in, open PowerShell with administrator privileges again, execute the command set-ExecutionPolicy Unrestricted, then run the script again. |
| 6<br>Update the shared Key<br>☐ - Done | WARNING<br>In our lab, we received an error 13801 in event viewer when the connection failed to start. Error 13801 indicates an invalid certificate. Working with the Azure team, we were able to get confirmation that there is actually no certificate used. It is the shared key that provides authentication and the error message is misleading. Further confusing the problem was that there is a place to put a certificate in the Azure virtual network configuration, but this is for point-to-site connections only. The VPN script you downloaded does set the shared key when it creates the virtual network. However, we found that we needed to explicitly copy and paste the shared key before starting the connection.<br><br>• On the Virtual Network dashboard, click **Manage Key**, then the **copy** icon.<br>  <br><br>• Open the **Routing and Remote Access** tool from Server Manager on the gateway server.<br>• Navigate to the newly created virtual network under **Network Interfaces**, then right-click **Properties**.<br>• On the **Security** tab, select the textbox under **Use preshared key for authentication**, and paste the clipboard contents into the box. |
| 6<br>Connect the VPN tunnel<br>☐ - Done | • On the Azure Virtual network page, click the **Connect** button. This will start the Azure gateway. You will get a message that the gateway has been started but it has failed to connect, and will retry the connection periodically.<br>• Open **Routing and Remote Access** from the server manager tools list.<br>• Navigate to the newly created virtual network, and click **Connect**<br>• After refreshing the RRAS console, you should see it has entered the connected state.<br>• Return to the virtual network dashboard. You should see a link between devices now. |

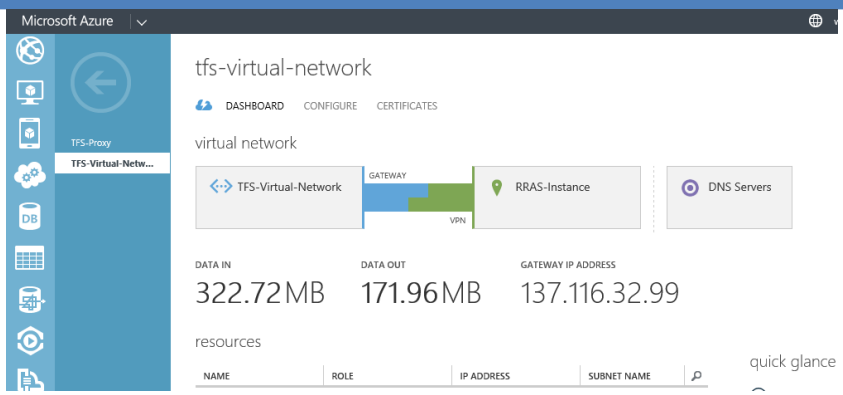
| Step | Instructions |
|---|---|
| |  |
| 7<br>Validate<br>⬚ - Done | • Add a virtual machine: **tfspoc-test (for testing purposes only)**<br>• Added local user name: **tfspocadmin**<br>• Set password: **Password@1**<br>• Set virtual network: **TFS-Virtual-Network**<br>• Set the virtual network subnet: **TFS-Farm-Seg**<br>• Click next to create the virtual machine.<br>• On the Virtual Machine dashboard, view IP Configuration.<br>• Ensure the assigned IP address is part of the virtual network segment **TFS-Farm-Seg.**<br>• Ensure Gateway is set to the IP Address listed on the virtual network dashboard.<br>• If you have ICMP opened on your on-premises lab, you can use Ping to verify connectivity.<br>• Connect to the server by clicking the Connect button to launch a Remote Desktop Connection. Open a command prompt.<br>• Ping the IP address of the DNS server and ensure it responds.<br>• Ping a known name on your internal network.<br>• You will not use this virtual machine again. Shut it down and delete it after completing your test. |

**Table 15 – Prepare Azure network environment**

# Azure Storage

## Database Storage Walkthrough

| Step | Instructions |
|---|---|
| 1<br>Create storage<br>☐ - Done | • Navigate to your Azure portal.<br>• Select **STORAGE** and **New**.<br>  <br>• Select **QUICK CREATE**.<br>• Specify a unique name for the **URL**: tfsazurepocstorage.<br>• Specify **LOCATION/AFFINITY GROUP** to match region which will contain the data-tier server: TFS-Affinity-Group (East US 2)<br>• Select **CREATE STORRAGE ACCOUNT** to create the storage. |
| 2<br>Configure storage<br>☐ - Done | • Optionally configure your storage to match your planned **storage replication**, **monitoring,** and **logging**. |

**Table 16 – Storage Setup**

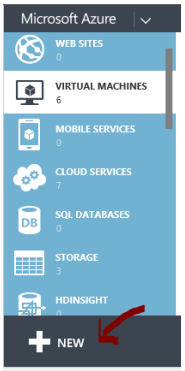
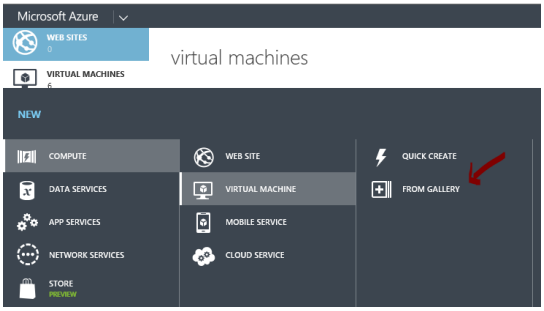
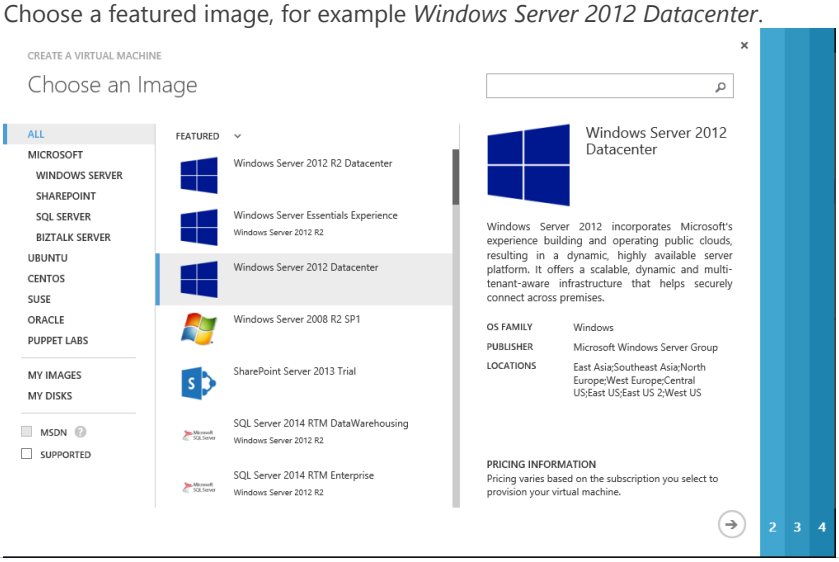
# Azure Servers

## Domain Controller

NOTE

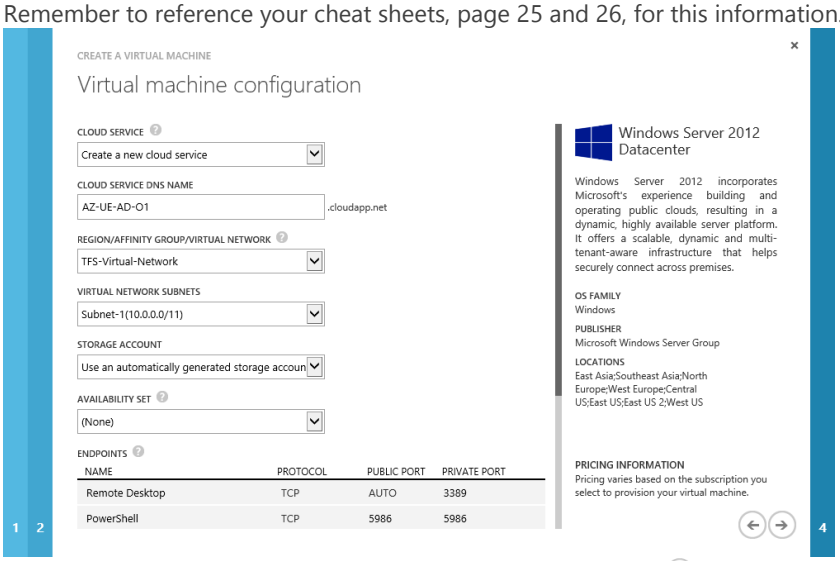
Refer to **Machine Planning**, page 25, for server information and **Credentials Planning**, page 26, for credential information needed for the setup. This information is unique to each environment and captured as part of deployment planning.

### Setup walkthrough

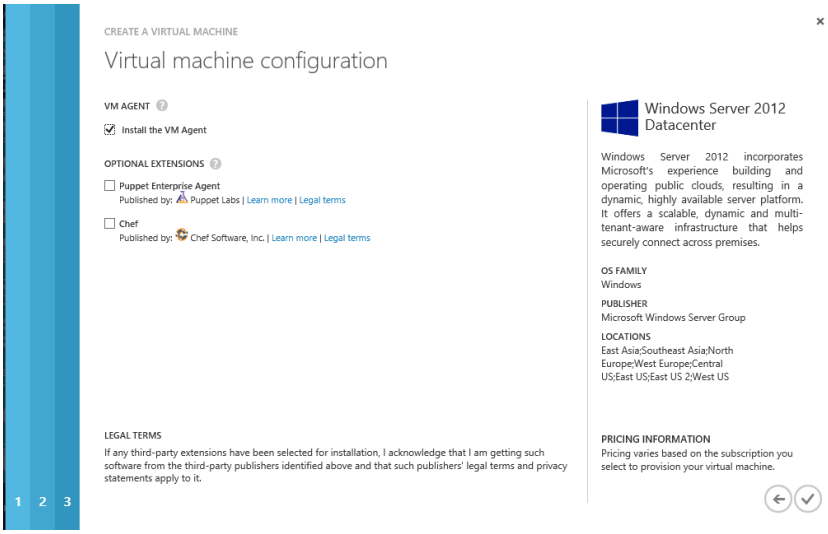
| Step | Instructions |
|---|---|
| 1<br>Create VM<br>☐ - Done | • Navigate to your Azure portal.<br>• Select **VIRTUAL MACHINES** and **New**.<br>  <br>• Select **FROM GALLERY**<br>  <br>• Choose a featured image, for example *Windows Server 2012 Datacenter*.<br>  |

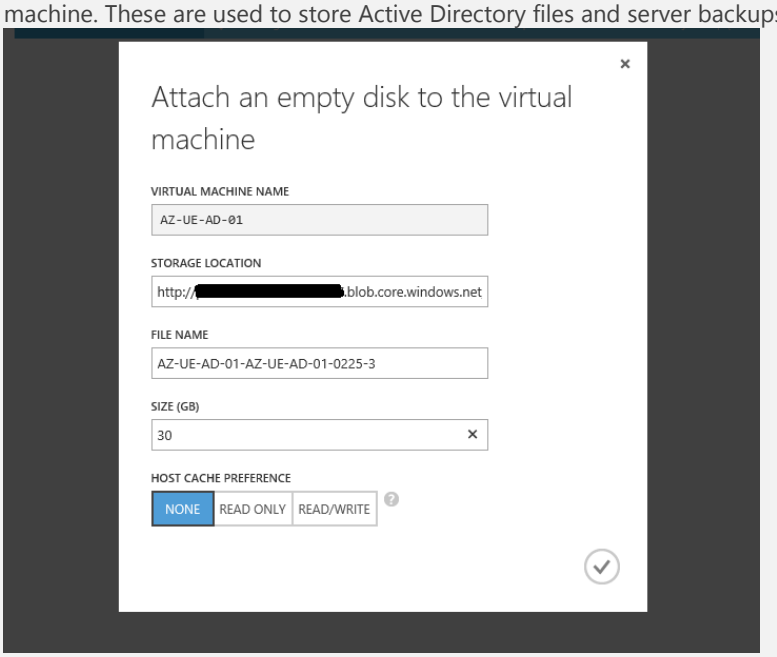
- Configure the image to reflect your design:


- Define the **VIRTUAL MACHINE NAME**: *AZ-UE-AD-O1*.
- Select the correct **SIZE:** *Small 1-Core*
- Select the **STORAGE ACCOUNT**: *automatic*.
- Specify the *TfsAzurePocUser* credentials for the **NEW USER NAME** for the Active Directory server. Remember to reference your cheat sheets, page 25 and 26, for this information.



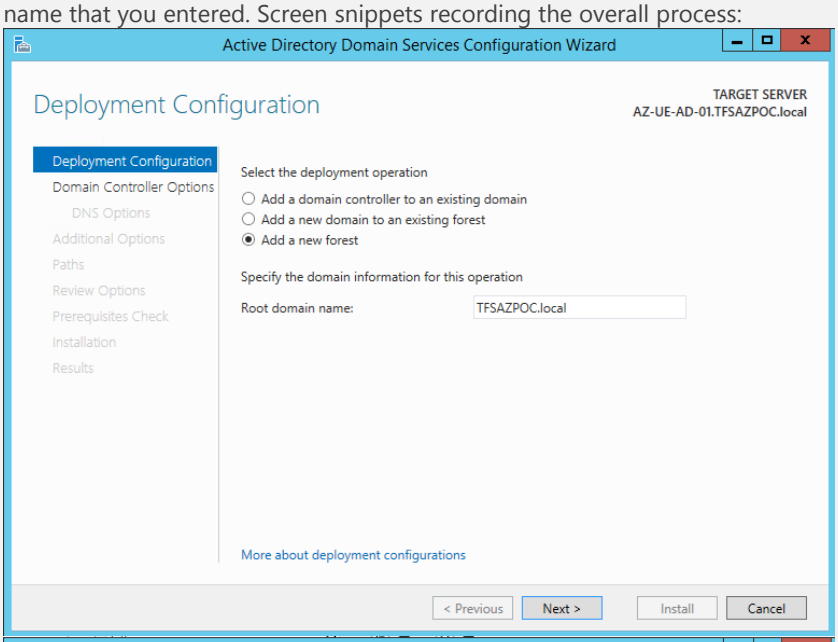

- Select the correct **AFFINITY GROUP**: *TFS-Virtual-Network* and
- Select the **VM AGENT** and .

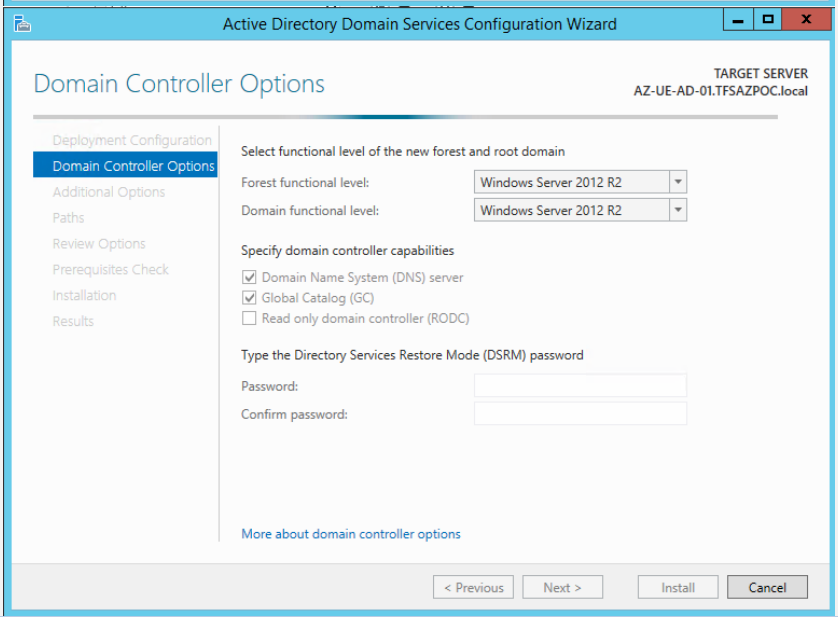


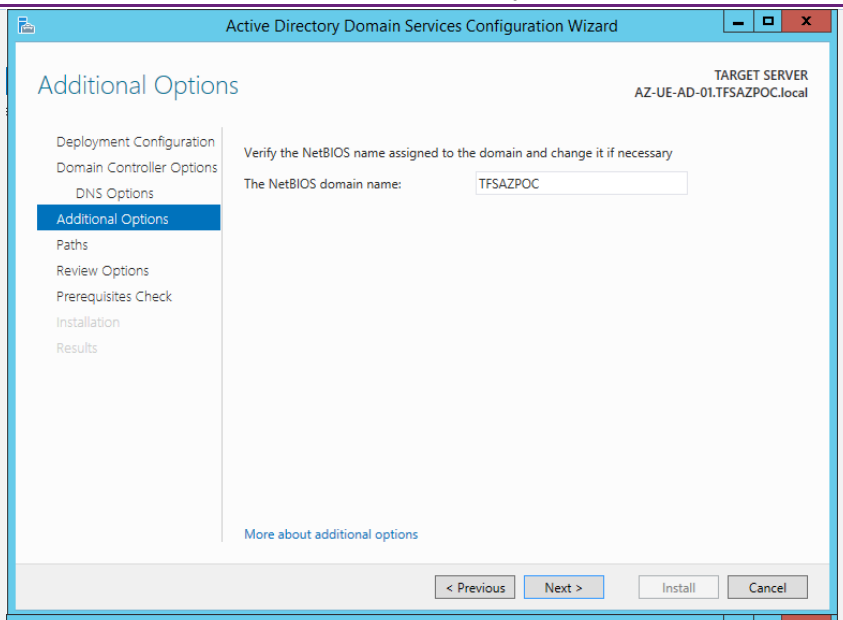
| | |
|---|---|
| 2<br>Attach disks<br>☐ - Done | • When the Virtual Machine is successfully provisioned, attach two empty 30 GB disks to the virtual machine. These are used to store Active Directory files and server backups.<br>  |
| 3<br>Connect to machine<br>☐ - Done | • Once you have successfully attached the disks, connect to the machine by selecting the VM name and clicking connect in the Azure management portal. Click **Connect** on the RDP connection dialog if a publisher warning is displayed.<br>• Then, in the Remote Desktop Connection dialog, enter the username and password assigned for the POC (see page 26) and click **Connect**.<br>• Click **Yes** if a certificate identity prompt is displayed. |
| 4<br>Install AD role<br>☐ - Done | • At this point, you need to install the Active Directory Domain Services (ADDS) role and upon completion, you should run a server backup.<br>• Detailed steps for how to do this can be found in the following Microsoft Azure online documentation: Install a new Active Directory forest in Microsoft Azure [73]<br>• For the purposes of this PoC, we used the domain *TFSAZUREPOC.local*, with the NETBIOS name *TFSAZUREPOC*. |

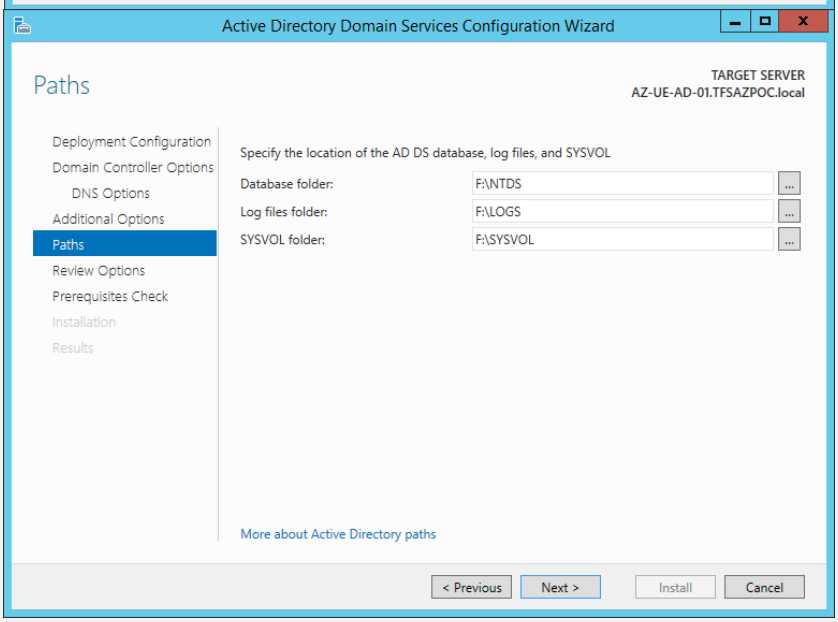
[73] http://www.windowsazure.com/en-us/documentation/articles/active-directory-new-forest-virtual-machine/#Step3

- Once you have successfully installed the Active Directory Domain Services role, the server will reboot and you will need to connect again via a Remote Desktop Connection. At this point, use the domain name that you entered. Screen snippets recording the overall process:





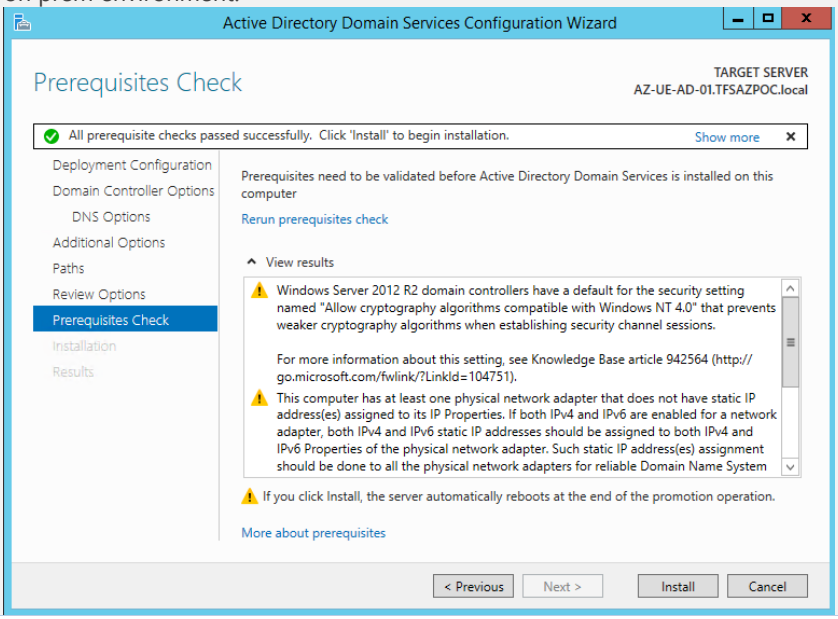

- Review options:
  - Configure this server as the first Active Directory domain controller in a new forest.
  - The new domain name is *TFSAZPOC.local*. This is also the name of the new forest.
  - The NetBIOS name of the domain: *TFSAZPOC*
  - Forest Functional Level: *Windows Server 2012 R2*
  - Domain Functional Level: *Windows Server 2012 R2*
- Additional Options:
  - DNS Server: *Yes*
  - Create DNS Delegation: *No*

  The DNS Server service will be configured on this computer. Expect to see the following messages:
  - This computer will be configured to use this DNS server as its preferred DNS server.
  - The password of the new domain Administrator will be the same as the password of the local Administrator of this computer.

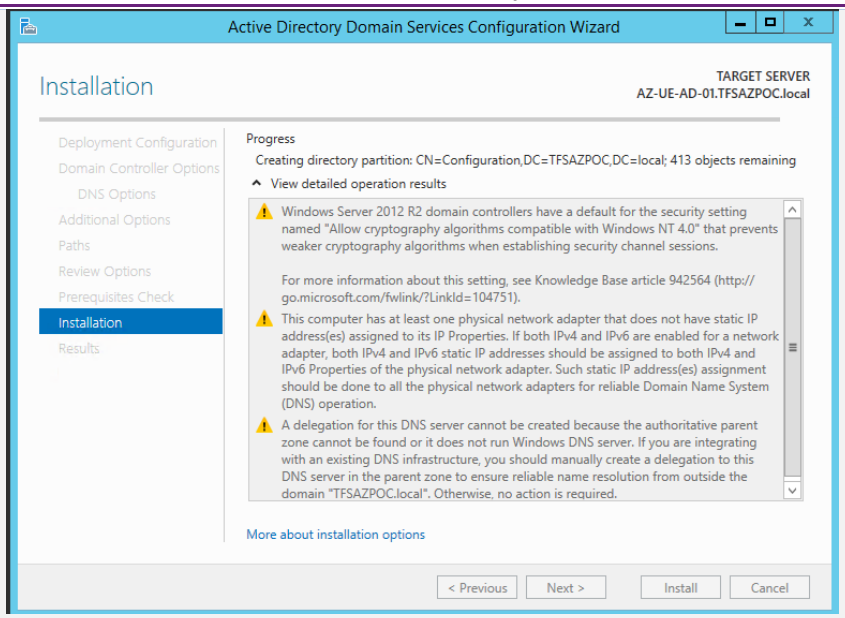
WARNING

Windows Server 2012 R2 domain controllers have a default for the security setting named "Allow cryptography algorithms compatible with Windows NT 4.0" that prevents weaker cryptography algorithms when establishing security channel sessions. You need to decide whether this default is inappropriate for your environment. See Knowledge Base article 942564 [74] for details.

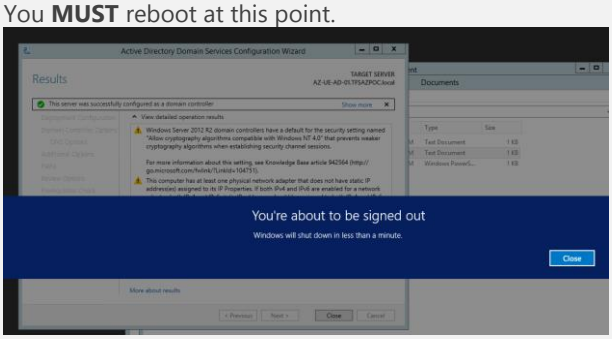
- This computer has at least one physical network adapter that does not have one or more static IP addresses assigned to its IP Properties. If both IPv4 and IPv6 are enabled for a network adapter, both IPv4 and IPv6 static IP addresses should be assigned to both IPv4 and IPv6 Properties of the physical network adapter. You should assign such static IP addresses to all the physical network adapters for reliable Domain Name System (DNS) operation.
- Because the domain is not a routable internet domain, it cannot be delegated as internet clients cannot reliably contact it. Delegation involves allowing zone transfer to another server, for instance allowing the azure computers to be reachable from clients connecting from the on-prem environment.
  This is a technical way of saying you cannot see azure machines from on prem because .Local is not resolved by a root server in the way .com is. To do this, you would need a forward reference in the on-prem environment, which we could have done by configuring a secondary zone for azure in the on prem environment.




[74] http://go.microsoft.com/fwlink/?LinkId=104751




- You **MUST** reboot at this point.




| | |
|---|---|
| 8<br>Validate installation<br>☐ - Done | • To validate the ADDS installation, launch a command prompt as Administrator and execute the following command:<br>  • dcdiag /c /v<br>• Verify that all the tests ran successfully, and refer to the instructions in the link below if you encounter any warnings or errors. See Validate the installation [75] for details. |
| 9<br>Backup domain controller<br>☐ - Done | • Following completion of the steps above, backup the VM following the instructions here: Backup the domain controller [76] |
| 10<br>Finalize<br>☐ - Done | • At the point, the Domain Controller is provisioned and functioning.<br>• You can now configure additional virtual machines to be domain-joined either on provisioning if using PowerShell or once provisioned by connecting to the virtual machine and manually joining the domain.<br>• See the following article for more information: Create and Upload a Management Certificate for Microsoft Azure [77] |

**Table 17 - Domain controller setup**

[75] http://www.windowsazure.com/en-us/documentation/articles/active-directory-new-forest-virtual-machine/#Step4

[76] http://www.windowsazure.com/en-us/documentation/articles/active-directory-new-forest-virtual-machine/#Step5

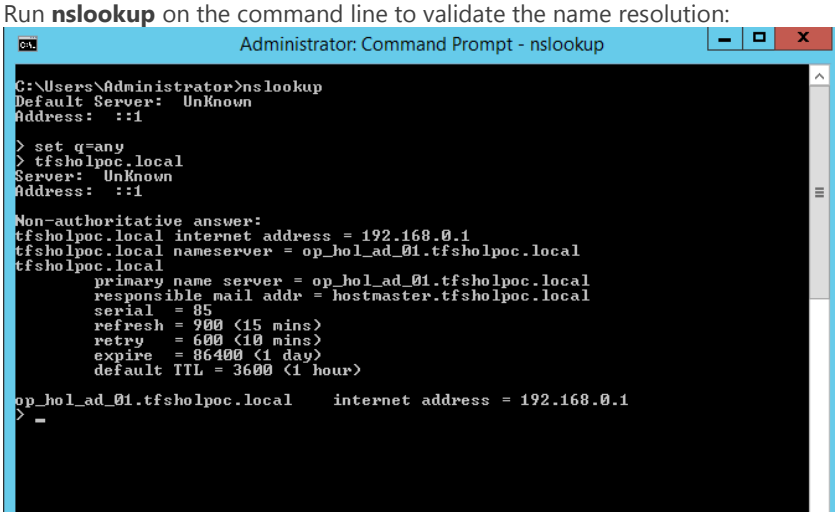
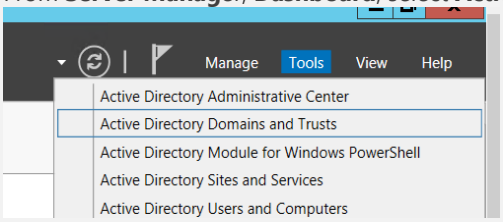
[77] http://msdn.microsoft.com/en-us/library/windowsazure/gg551722.aspx

## One-way trust to on-premises AD walkthrough

NOTE

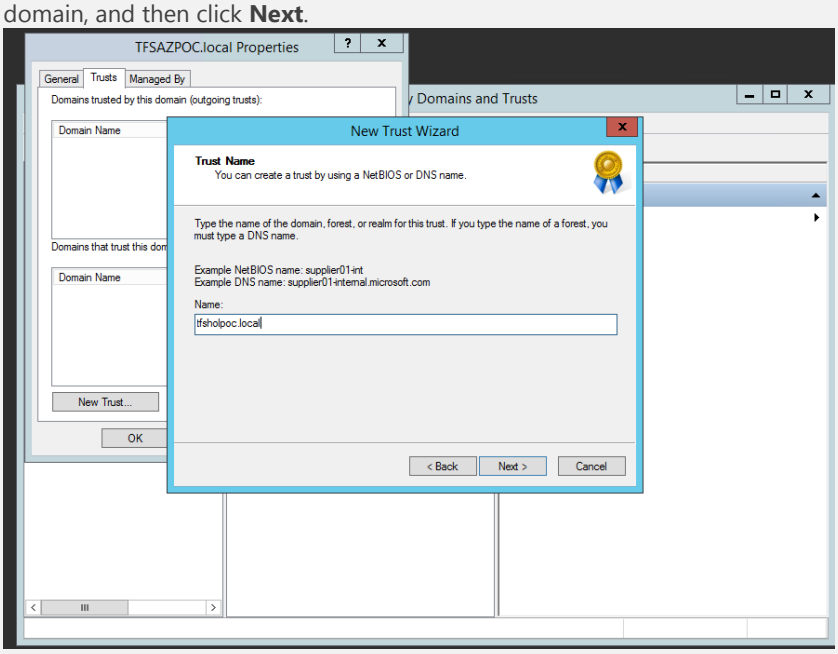
You only need this step if you are using on-premises Active Directory for the user credentials.

A one-way trust will allow you to manage user accounts in the on-premises Active Directory rather than the Azure directory. This provides additional protection for the corporate network, while also centralizing management of user accounts. Configuring a one-way trust is optional but recommended.

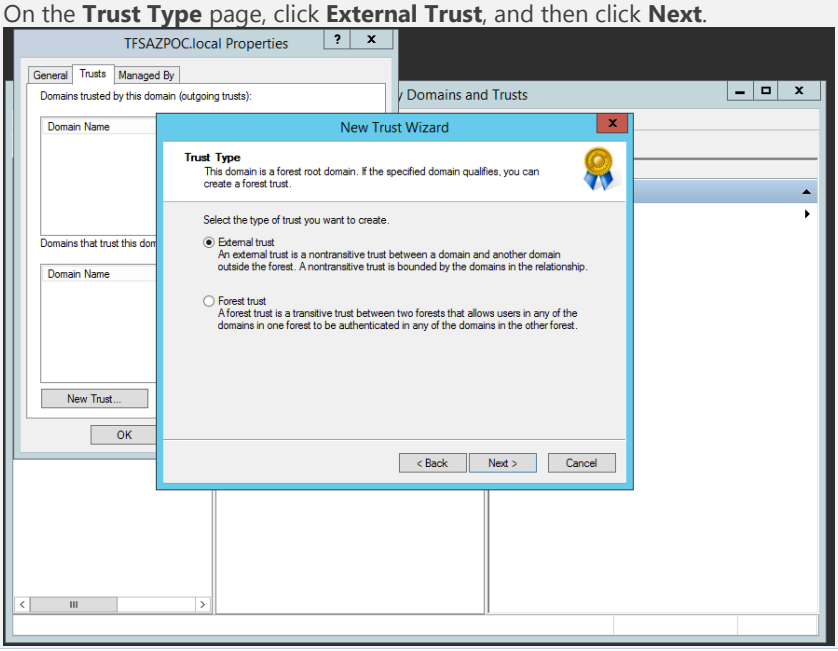
| Step | Instructions |
|---|---|
| 1<br>NS Lookup<br>☐ - Done | • Logon to the Azure domain controller as domain administrator.<br>• Run **nslookup** on the command line to validate the name resolution:<br>  |
| 2<br>Create Trust<br>☐ - Done | • You can find detailed steps for this setup in the online documentation: Create an External Trust [78].<br>• From **Server Manager**, **Dashboard**, select **Active Directory Domains and Trusts**.<br>  <br>• In the console tree, right-click the domain node for the domain with which you want to establish a trust, and then click **Properties**.<br>• On the **Trusts** tab, click the **New Trust**, and then click **Next**. |

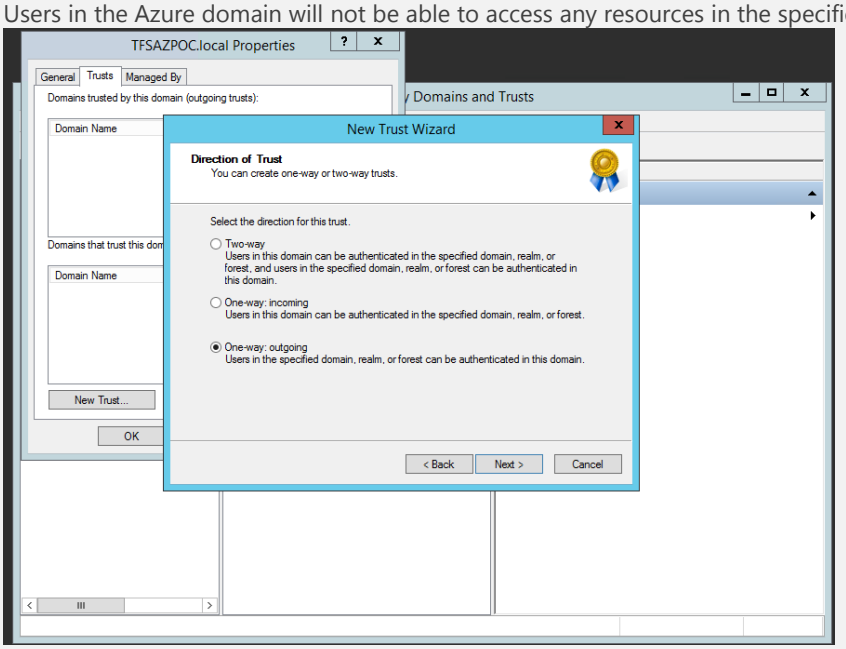
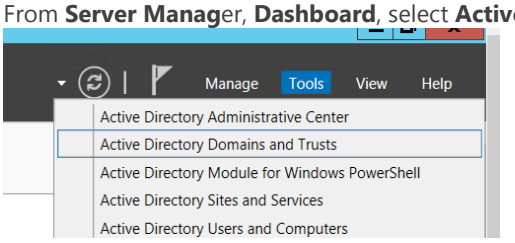
[78] http://technet.microsoft.com/en-us/library/cc771580.aspx

- On the **Trust Name** page, type the Domain Name System (DNS) name (or NetBIOS name) of the domain, and then click **Next**.



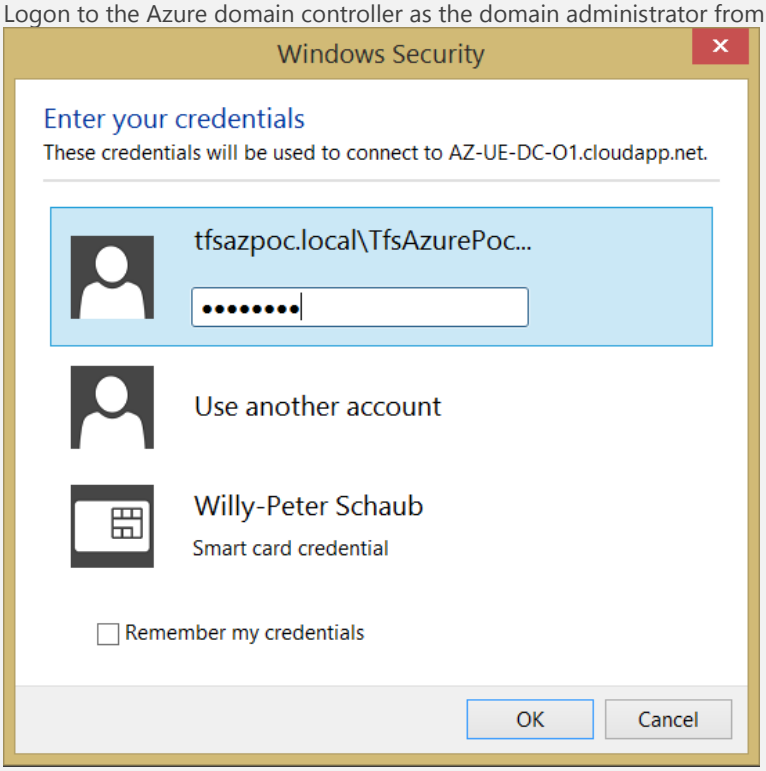

- On the **Trust Type** page, click **External Trust**, and then click **Next**.



| | |
|---|---|
| | • Create a one-way, **outgoing** external trust.<br>Users in the Azure domain will not be able to access any resources in the specified domain.<br>  <br><br>• Select **both this domain and the specified domain** in next dialog.<br>• Continue to follow the instructions in the wizard and remember the password you specify for the trust. |
| 3<br>Verify Trust<br>☐ - Done | • Logon to the on-premises domain controller.<br>• You can find detailed steps for this setup in the online documentation: Verifying a trust [79].<br>• From **Server Manage**r, **Dashboard**, select **Active Directory Domains and Trusts**.<br>  <br><br>• In the console tree, right-click the domain containing the trust you want to verify, and then click **Properties**.<br>• On the Trusts tab, under **Domains that trust this domain** (incoming trusts), click the trust to be verified (the Azure region), and then click Properties.<br>• Click **Validate**.<br>• Click **Yes**, validate the incoming trust, and then click **OK**. |
| 4<br>Test Trust<br>☐ - Done | • Try to secure an object in your Azure environment using accounts from the on-premises environment. For the POC, we added on-premises accounts to the local administrators group so we could use then to log in.<br>• To do this, navigate to the local users and groups, and then add accounts to the administrators group from your on-premises domain. |

[79] http://technet.microsoft.com/en-us/library/cc753821.aspx

- Logon to the Azure domain controller as the domain administrator from the on-premises domain.




- If the logon works, you have validated the one-way trust.

**Table 18 - Domain controller setup**

# Data Tier (DT) Server

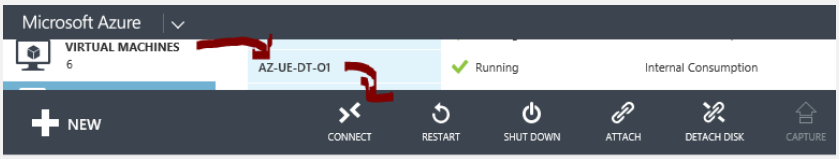
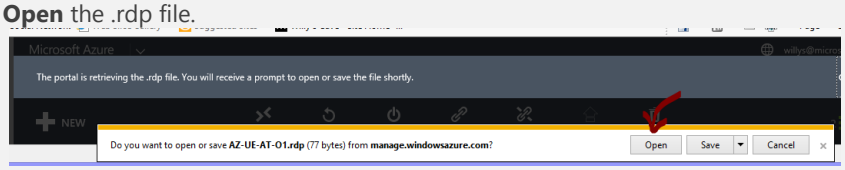
NOTE

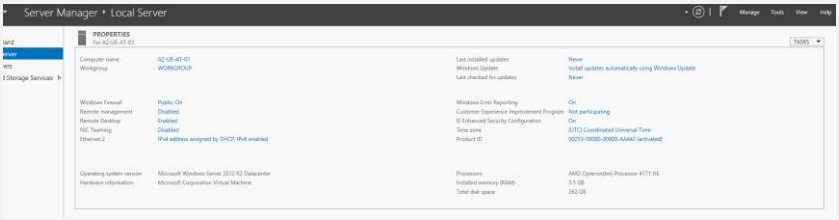
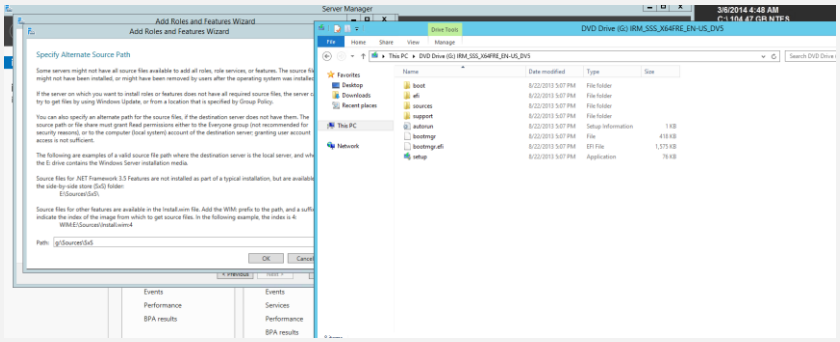
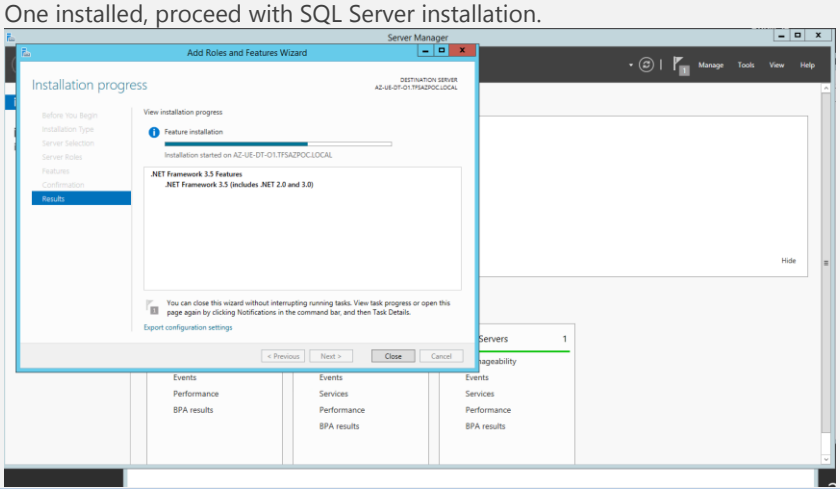
Refer to **Machine Planning**, page 25, for server information and **Credentials Planning**, page 26, for credential information needed for the setup. This information is unique to each environment and captured as part of deployment planning.

NOTE

To minimize the size of these walkthroughs, we are not repeating screenshots that are similar to the previous walkthrough steps, such as the Azure VM setup.

## Setup walkthrough

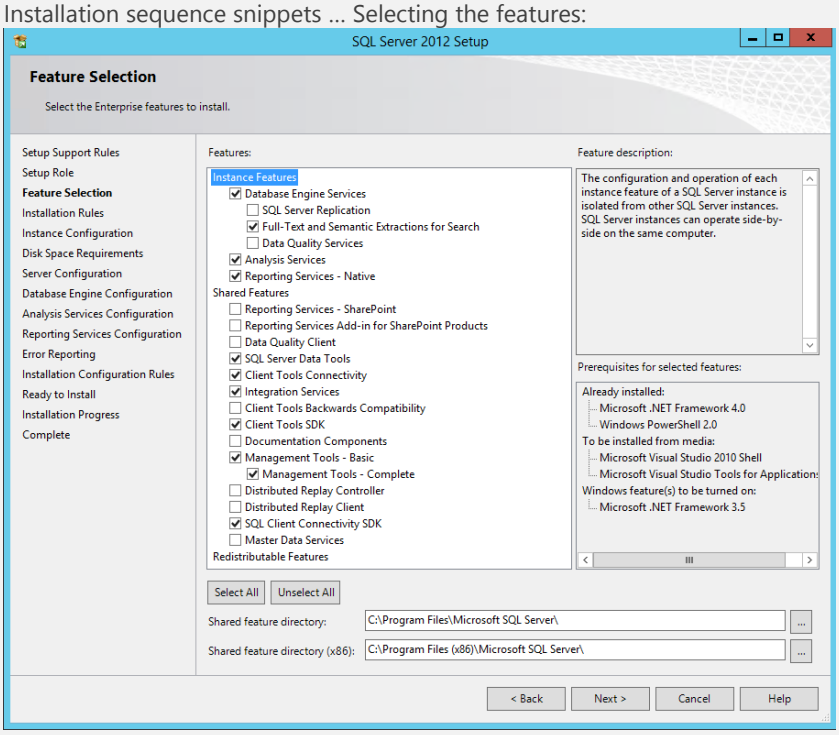
| Step | Instructions |
|---|---|
| 1<br>Create VM<br>[ ] - Done | • Navigate to your Azure portal.<br>• Select **VIRTUAL MACHINES** and **New**.<br>• Select **FROM GALLERY.**<br>• Choose a featured image, for example *Windows Server 2012 Datacenter*.<br>• Configure the image to reflect your design:<br>  <br><br>• Define the **VIRTUAL MACHINE NAME**: *AZ-UE-DT-O1*.<br>• Select the correct **SIZE:** *Large 4-Core*<br>• Specify the *TfsAzurePocUser* credentials for the **NEW USER NAME** for the data-tier server. Remember to reference your cheat sheets, page 25 and 26, for the information.<br>• Select the **STORAGE ACCOUNT**: *tfsazurepocstorage*<br>• Select the correct **AFFINITY GROUP**: *TFS-Virtual-Network* and<br>• Select the **VM AGENT** and . |
| 2<br>Login to virtual machine<br>[ ] - Done | • Once the virtual machine is provisioned and running, select the new data-tier VM instance and **CONNECT**.<br>  <br><br>• **Open** the .rdp file.<br>  <br><br>• **Connect** to the virtual machine.<br>• Enter the local admin credentials you specified when creating the virtual machine.<br>• Select **Yes** to connect. |
| 3<br>Optional PING test | • Once logged on, ping the Azure POC domain controller by IP and Name. |

| Step | Instructions |
|---|---|
| ☐ - Done | |
| 4<br>Disable IE Enhanced Security<br>☐ - Done | • Select **Server Manager**, **Local Server** and disable the **Internet Explorer Enhanced Security** to be able to download software from the internet.<br>  |
| 5<br>Download software<br>☐ - Done | • Download SQL Server 2012 SP1 from your installation directory, network share, or other location, for example from MSDN subscriptions: MSDN SQL Family [80]. |
| 6<br>Install and configure software | • Install and configure SQL Server.<br>• SQL Server 2012 needs .NET 3.5<br>  • Attach the ISO of the Windows Server 2012 to the machine.<br>  • Use the **Add Features** to find the install on Windows Server 2012.<br>  <br>  • One installed, proceed with SQL Server installation.<br>  |

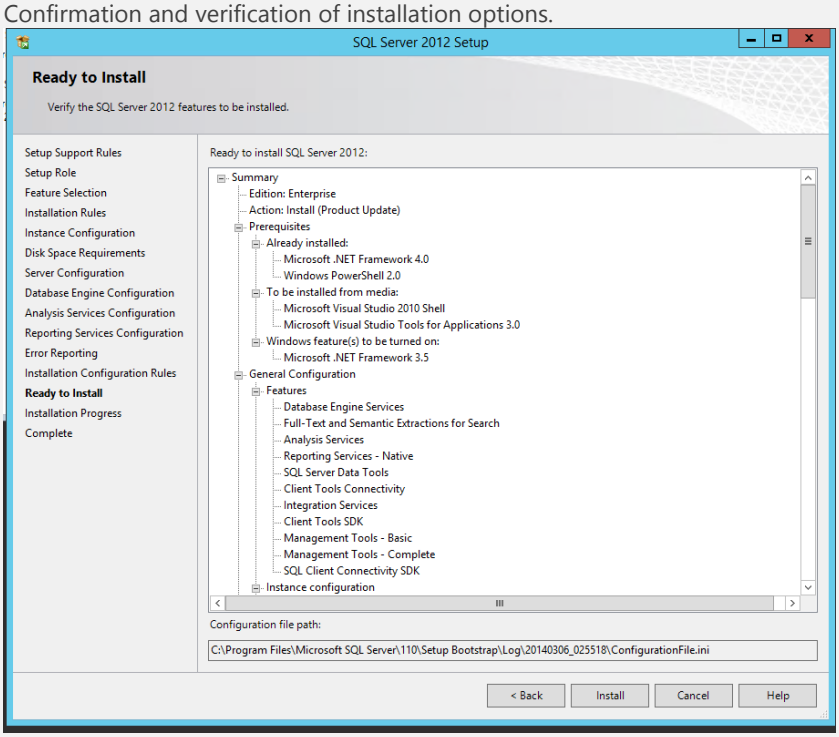
[80] https://msdn.microsoft.com/en-us/subscriptions/securedownloads/hh442898#searchTerm=&ProductFamilyId=461&Languages=en&PageSize=50&PageIndex=0

| Step | Instructions |
|---|---|

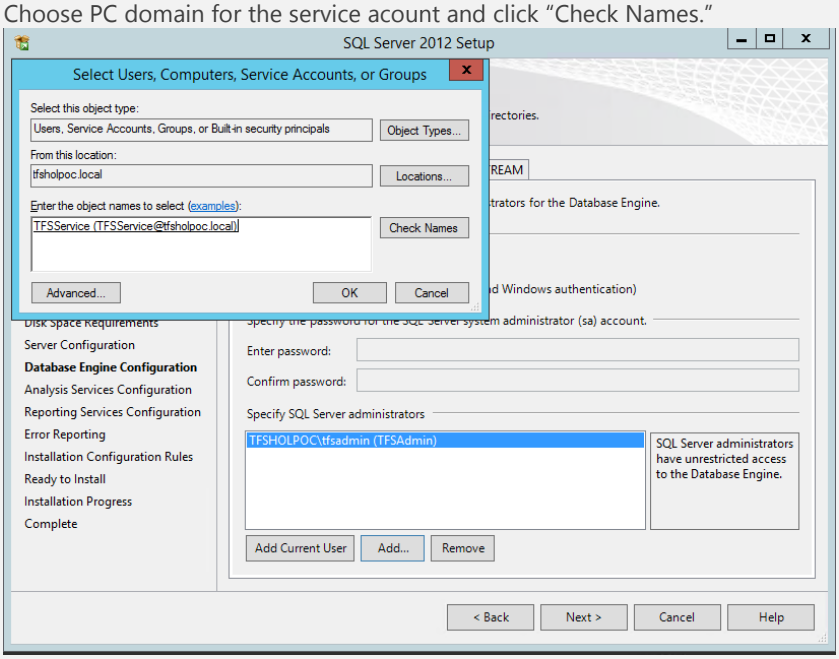
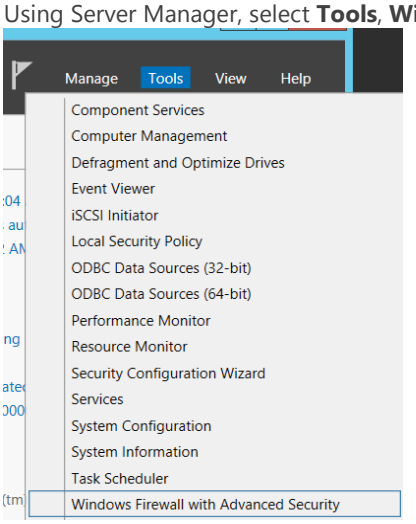
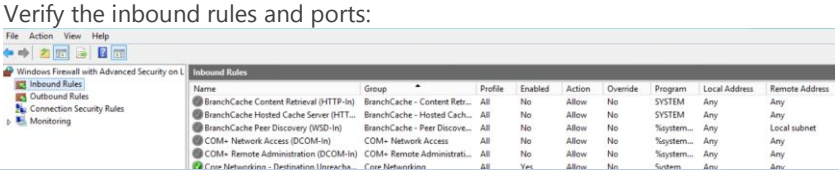
- Installation sequence snippets … Selecting the features:




- Confirmation and verification of installation options.



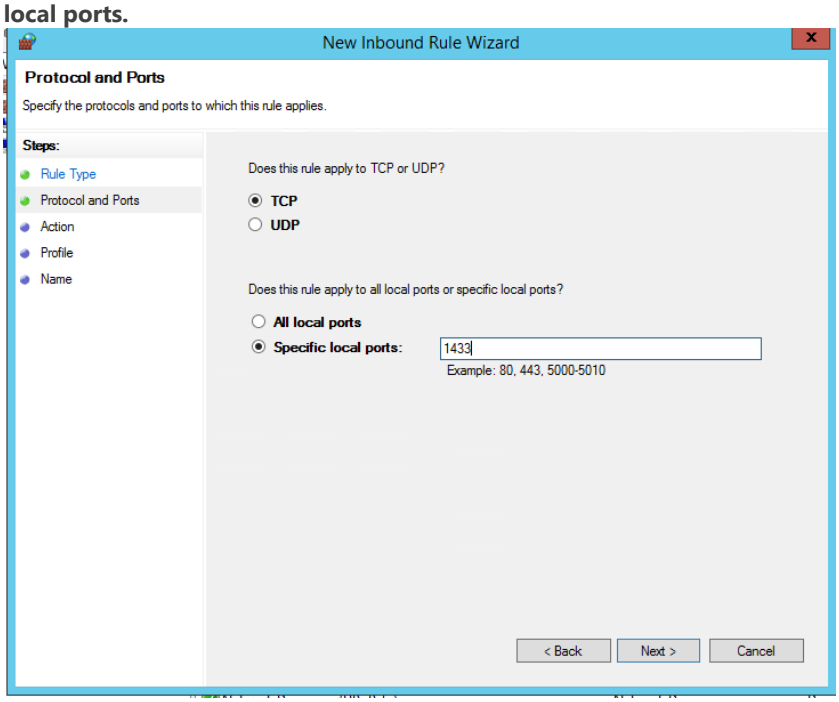

- Specify the SQL Server administrator using the TFS service account.

| Step | Instructions |
|---|---|
| | • Choose PC domain for the service acount and click "Check Names."<br>  <br>• Select **Next** and complete the installation. |
| 7<br>Verify Firewall<br>⬚ - Done | • Review TFS Default Network Settings [81] and verify that the required firewall ports were opened during the installation.<br>• Using Server Manager, select **Tools**, **Windows Firewall with Advanced Security:**<br>  <br>• Verify the inbound rules and ports:<br>  |

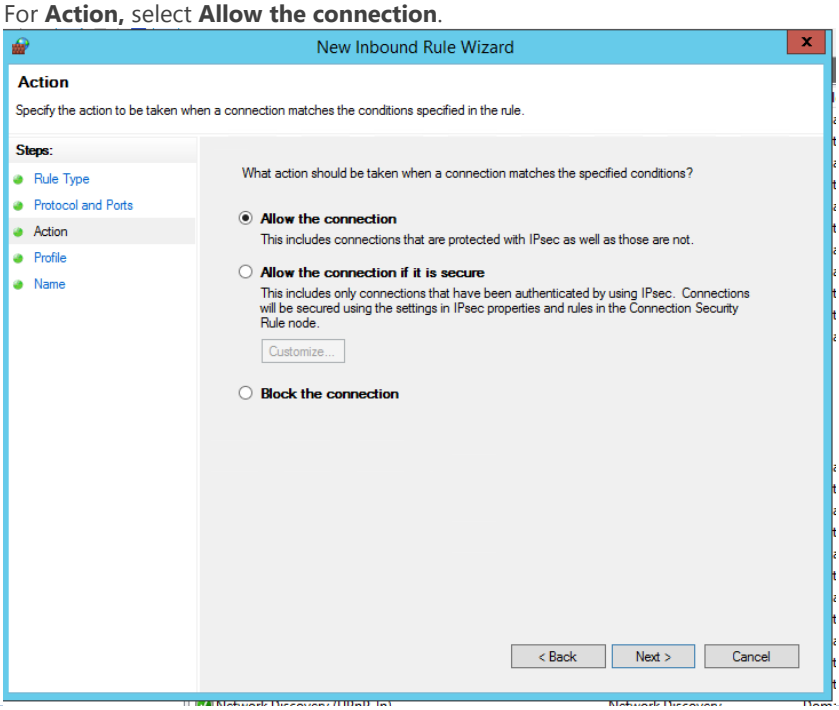
[81] http://msdn.microsoft.com/en-us/library/ms252473.aspx#Default

| Step | Instructions |
|---|---|

- If the ports are not open, use the **New Inbound Rule Wizard** and specify **TCP** rule type and **Specific local ports.**

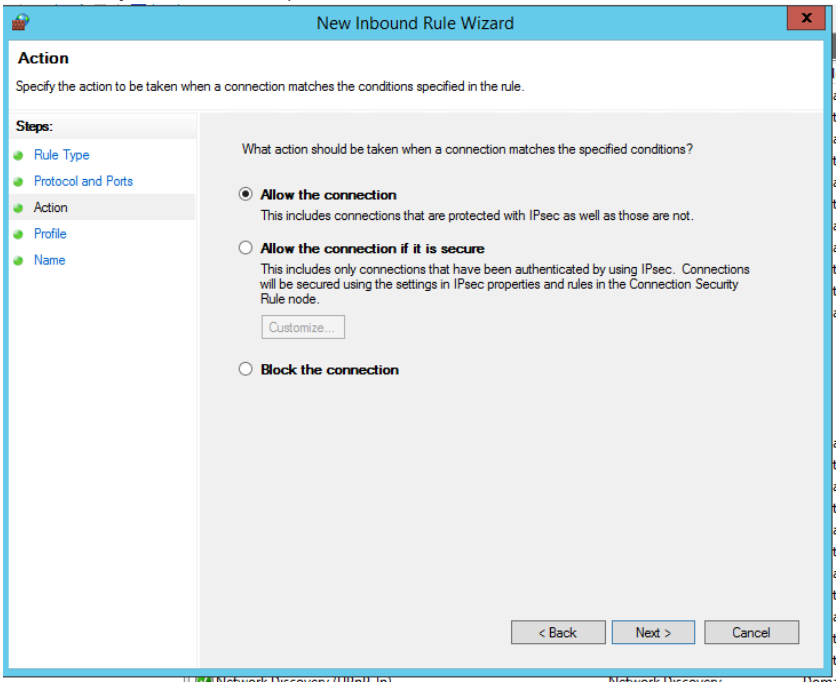



- For **Action,** select **Allow the connection**.

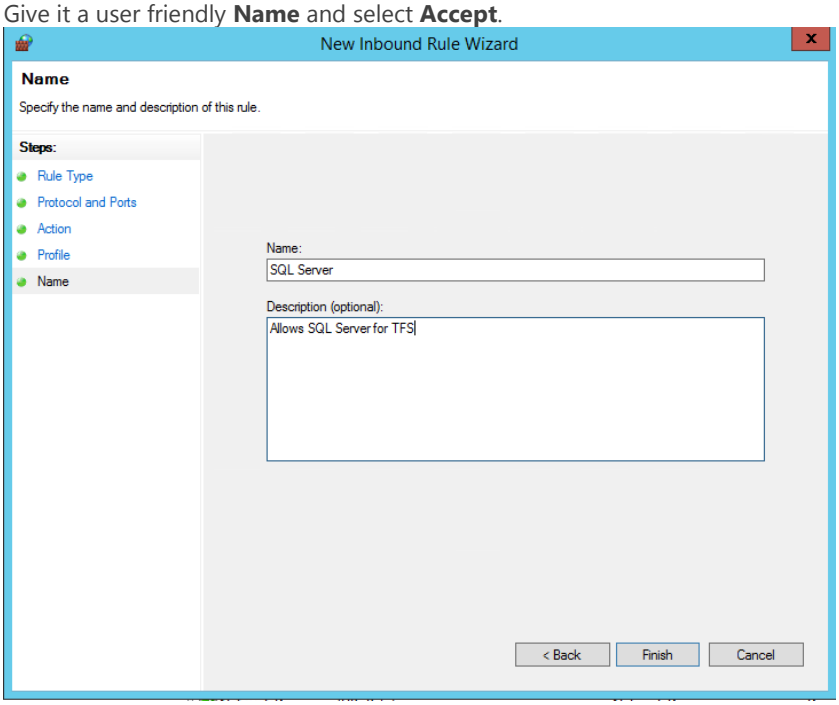


| Step | Instructions |
|---|---|

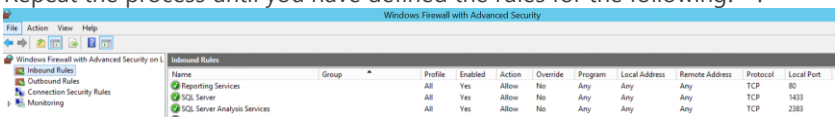
- For **Profile,** select **Domain**, **Private** and **Public**.




- Give it a user friendly **Name** and select **Accept**.




- Repeat the process until you have defined the rules for the following: [82]:




  - Reporting Services port: 80
  - SQL Server port: 1433

[82] Ports used in this walkthrough are specific to the POC and SQL Server used.

| Step | Instructions |
|---|---|
| | • SQL Server Analysis Services port: 2383 |

**Table 19 – Data Tier server setup**

# Application Tier (AT) Server

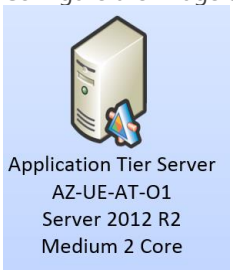
NOTE

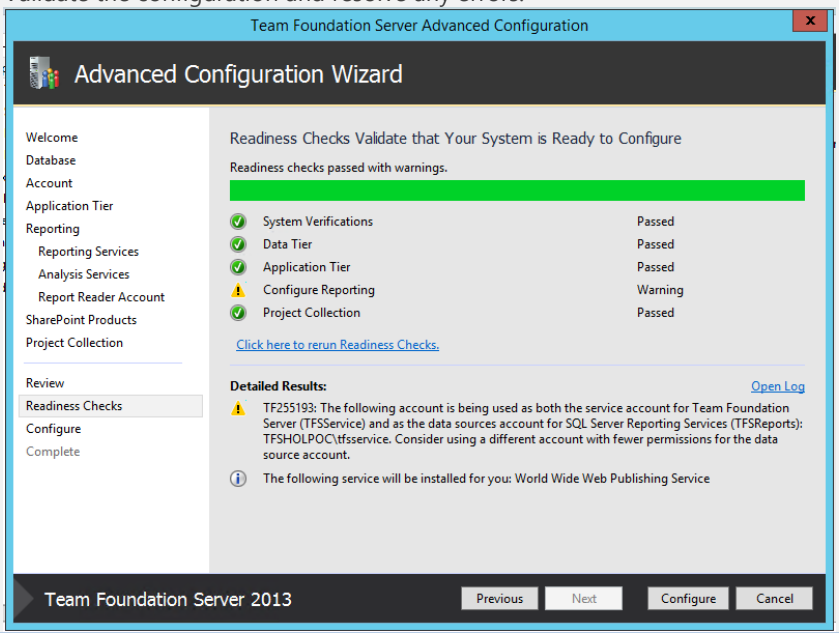
Refer to **Machine Planning**, page 25, for server information and **Credentials Planning**, page 26, for credential information needed for the setup. This information is unique to each environment and captured as part of deployment planning.

NOTE

To minimize the size of these walkthroughs, we are not repeating screenshots that are similar to the previous walkthrough steps, for example Azure VM setup.

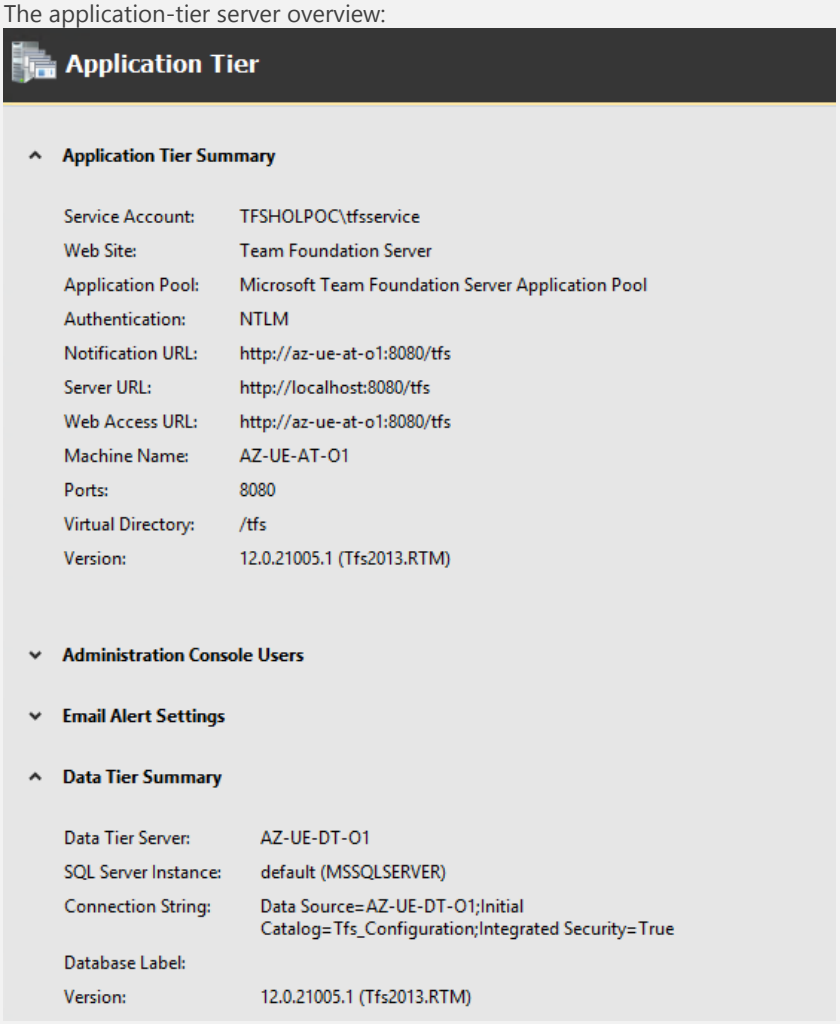
## Setup walkthrough

| Step | Instructions |
|---|---|
| 1<br>Create VM<br>⬚ - Done | • Navigate to your Azure portal.<br>• Select **VIRTUAL MACHINES** and **New**.<br>• Select **FROM GALLERY.**<br>• Choose a featured image, for example Windows Server 2012 Datacenter.<br>• Configure the image to reflect your design:<br>  <br><br>• Define the **VIRTUAL MACHINE NAME**: *AZ-UE-AT-O1*.<br>• Select the correct **SIZE:** *Medium 2-Core*<br>• Specify the *TfsAzurePocUser* credentials for the **NEW USER NAME** for the application-tier server. Remember to reference your cheat sheets, page 25 and 26, for the information.<br>Select the **STORAGE ACCOUNT**: *automatic*<br>• Select the correct **AFFINITY GROUP**: *TFS-Virtual-Network* and ➔<br>• Select the **VM AGENT** and ✓. |
| 2<br>Login to virtual machine<br>⬚ - Done | • Once virtual machine has been provisioned and running, select the new AT VM instance and **CONNECT**.<br>• **Open** the .rdp file.<br>• **Connect** to the virtual machine.<br>• Enter the local admin credentials you specified when creating the virtual machine.<br>• Select **Yes** to connect. |
| 3<br>Optional PING test<br>⬚ - Done | • Once logged on, ping the Azure POC domain controller by IP and Name. |
| 4<br>Disable IE Enhanced Security<br>⬚ - Done | • Select **Server Manager**, **Local Server** and disable **Internet Explorer Enhanced Security** to be able to download software from the internet. |

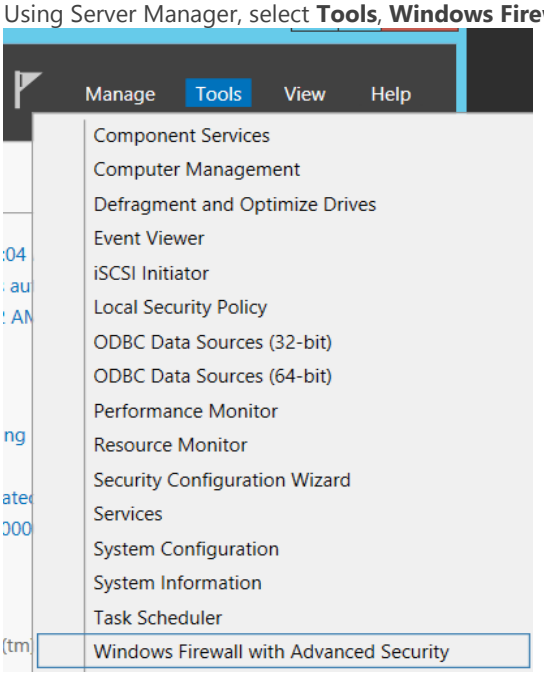
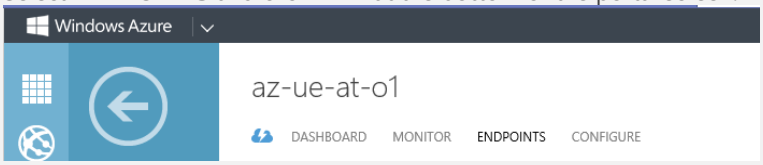
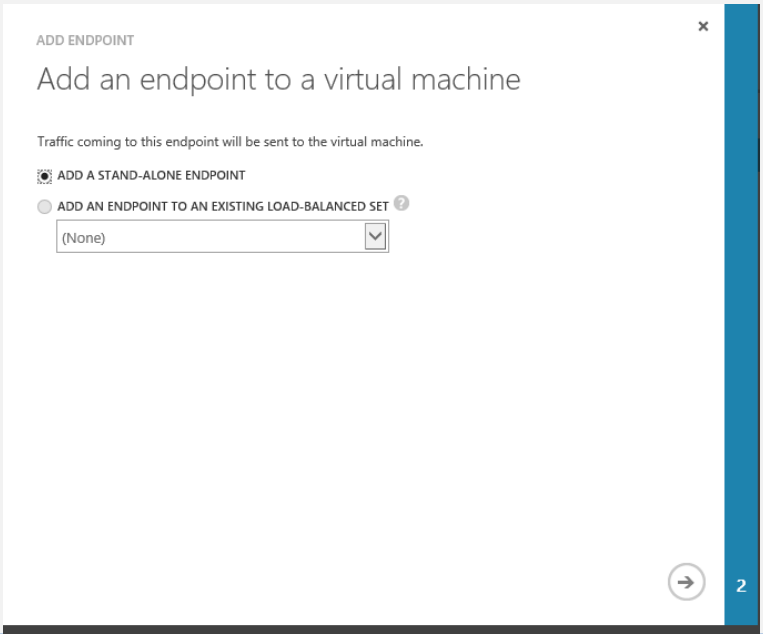
| Step | Instructions |
|---|---|
| 5<br>Download software<br>☐ - Done | • Download the Team Foundation Server software, for example from MSDN subscriptions: MSDN TFS Family [83]. |
| 6<br>Install and configure software<br>☐ - Done | • Install and configure Team Foundation Server. We recommend using the Advanced Wizard for configuration for maximum control over your environment's settings.<br>• See Team Foundation Server install guide [84] for details.<br>• Select the data-tier server as the **SQL Server Instance.**<br>• Specify the TFS Service account.<br>Remember to reference your cheat sheets, page 25 and 26, for the information.<br>• Validate the configuration and resolve any errors:<br>  |

[83] https://msdn.microsoft.com/en-us/subscriptions/securedownloads/hh442898#searchTerm=Visual%20Studio%20Team%20Foundation%20Server%202013&ProductFamilyId=0&Languages=en&PageSize=10&PageIndex=0&FileId=0

[84] http://msdn.microsoft.com/en-us/library/dd631902.aspx

| Step | Instructions |
|---|---|
| | • The application-tier server overview:<br>  <br> |
| 7<br>Verify Firewall<br>☐ - Done | • Review TFS Default Network Settings [85] and verify that the required firewall ports were opened during the installation. |

[85] http://msdn.microsoft.com/en-us/library/ms252473.aspx#Default

| Step | Instructions |
|---|---|
| | • Using Server Manager, select **Tools**, **Windows Firewall with Advanced Security:**<br>  <br>• Verify the inbound rules and ports<br>  |
| 8<br>Add Endpoint<br>⬚ - Done | • On your Azure portal, select the application-tier virtual machine.<br>• Select **ENDPOINTS** and click  at the bottom of the portal screen:<br>  <br>• Select **ADD A STANDALONE ENDPOINT:**<br>  |

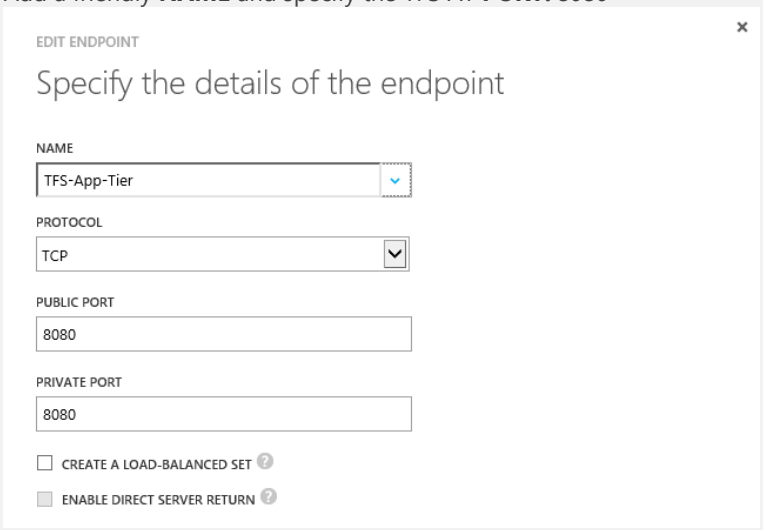
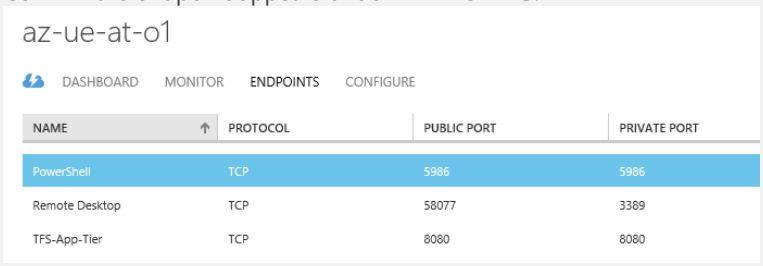
| Step | Instructions |
|---|---|
| | • Add a friendly **NAME** and specify the TFS AT **PORT:** *8080*<br>  <br>• Confirm the endpoint appears under **ENDPOINTS**.<br>  |
| 9<br>Validate AT Server<br>☐ - Done | • Connect to TFS from outside Azure using your browser: *http://az-ue-at-o1.cloudapp.net:8080/tfs/DefaultCollection* using the administrator account to validate connectivity and TFS.<br>• Connect to TFS from outside Azure using Visual Studio Team Explorer.<br>• Optionally validate the AT and DT servers using the functional test plan. See page 73 for details. |

**Table 20 – Application Tier server setup**

# Build Server (BS) Walkthrough

NOTE

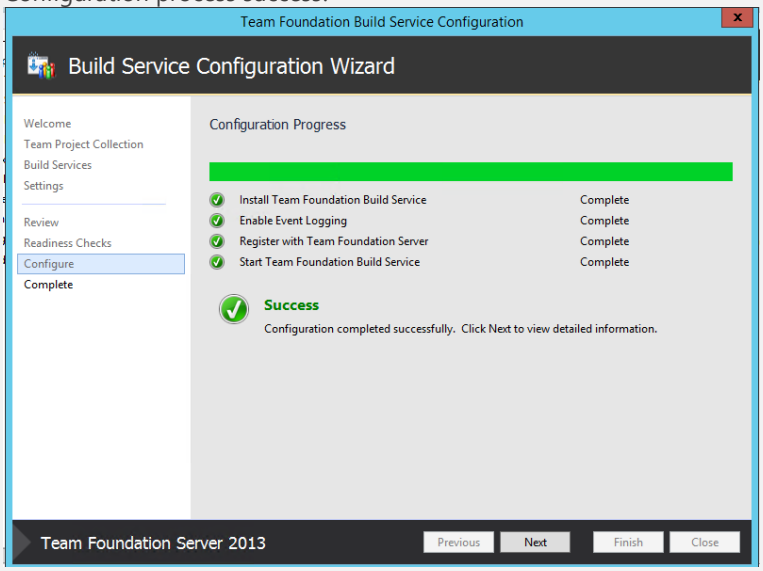
Refer to **Machine Planning**, page 25, for server information and **Credentials Planning**, page 26, for credential information needed for the setup. This information is unique to each environment and captured as part of deployment planning.

NOTE

To minimize the size of these walkthroughs, we are not repeating screenshots that are similar to the previous walkthrough steps, for example Azure VM setup.

## Setup walkthrough

| Step | Instructions |
|---|---|
| 1<br>Create VM<br>☐ - Done | • Navigate to your Azure portal.<br>• Select **VIRTUAL MACHINES** and **New**.<br>• Select **FROM GALLERY.**<br>• Choose a featured image, for example Windows Server 2012 Datacenter.<br>• Configure the image to reflect your design:<br>  <br><br>• Define the **VIRTUAL MACHINE NAME**: *AZ-UE-BS-O1*.<br>• Select the correct **SIZE:** *Medium 2-Core*<br>• Specify the *TfsAzurePocUser* credentials for the **NEW USER NAME** for the build server. Remember to reference your cheat sheets, page 25 and 26, for the information.<br>Select the **STORAGE ACCOUNT**: *automatic*<br>• Select the correct AFFINITY GROUP: TFS-Virtual-Network and  <br>• Select the VM AGENT and  . |
| 2<br>Login to virtual machine<br>☐ - Done | • Once the virtual machine is provisioned and running, select the new build server VM instance and **CONNECT**.<br>• **Open** the .rdp file.<br>• **Connect** to the virtual machine.<br>• Enter the local admin credentials you specified when creating the virtual machine.<br>• Select **Yes** to connect.<br>• Wait while the RDP session is established. |
| 3<br>Optional PING test<br>☐ - Done | • Once logged on, ping the Azure POC domain controller by IP and Name. |

| Step | Instructions |
|---|---|
| 4<br>Disable IE Enhanced Security<br>☐ - Done | • Select **Server Manager**, **Local Server** and disable **Internet Explorer Enhanced Security** to be able to download software from the internet. |
| 5<br>Download software<br>☐ - Done | • Download the Team Foundation Server software, for example from MSDN subscriptions: MSDN TFS Family [86]. |
| 6<br>Install and configure software | • Install and configure Team Foundation Server.<br>• Use the **Build** wizard.<br>See Set up Team Foundation Build Service and Deploy and configure a build server [87] [88] for details.<br>• Configuration process success!<br>  |
| 7<br>Verify Firewall<br>☐ - Done | • Review TFS Default Network Settings [89]and verify that the required firewall ports were opened during the installation. |

[86] https://msdn.microsoft.com/en-us/subscriptions/securedownloads/hh442898#searchTerm=Visual%20Studio%20Team%20Foundation%20Server%202013&ProductFamilyId=0&Languages=en&PageSize=10&PageIndex=0&FileId=0

[87] http://msdn.microsoft.com/en-us/library/ms181712.aspx

[88] http://msdn.microsoft.com/en-us/library/dd631902.aspx

[89] http://msdn.microsoft.com/en-us/library/ms252473.aspx#Default

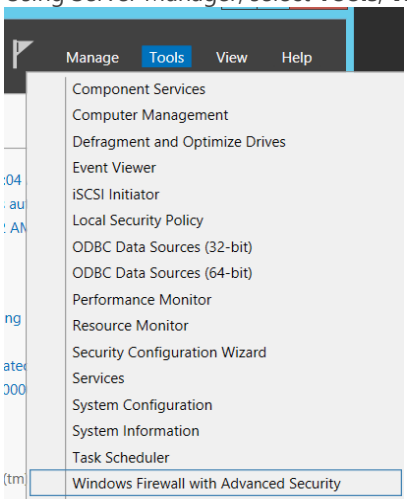
| Step | Instructions |
|---|---|
| | • Using Server Manager, select **Tools**, **Windows Firewall with Advanced Security:**<br>  <br>• Verify the inbound rules and ports:<br>  |

**Table 21 – Build server setup**

# Test Plan

## Understanding the domains

We discussed access and issues only to realize that users were trying to access the environment through the incorrect channel. We identified this as a definite **GOTCHA** , which warrants a brief explanation.

When we go back to the proposed companion proof of concept we note that the on-premises domain (TFSHOLPOC) simulates the corporate domain and would contain all users and machine accounts for the organization. The domain on Azure (TFSAZPOC) contains specialized administration accounts and the machine accounts for the TFS environment.

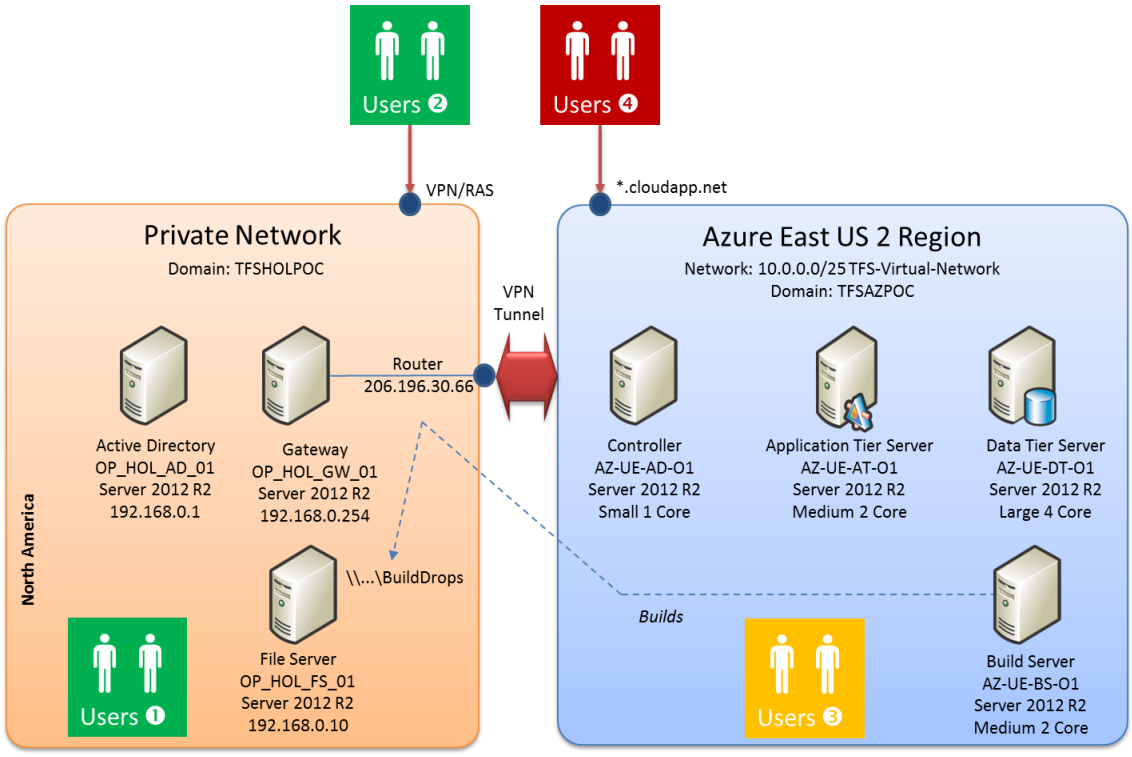



**Figure 11 - Understanding the POC domains**

Users would therefore access the TFS on Azure IaaS environment by doing one of the following:

1. Logging onto their machine on the corporate **on-premises** network, and then accessing the TFS on Azure IaaS environment.
2. Access the corporate **on-premises** network, by **VPN**/**RAS** connectivity, as if they were on-premises, and then accessing the TFS on Azure IaaS environment.
3. Logging onto a machine on the **Azure network**, typically for administration and maintenance only.
4. Connecting to the TFS on Azure IaaS environment from anywhere in the world, but not authenticated by the on-premises (TFSHOLPOC) or Azure (TFSAZPOC) domain.

> **WARNING**
>
> This is NOT a default or recommended scenario and not supported at this stage of the walkthrough.
> See **TFS on Azure IaaS v1.4 Supplement - Extend access beyond the on-premises and Azure domain**, included in the guidance downloads for details.

# Companion POC Functional Testing

GOAL

The POC functional test plan identifies what should to be tested as a bare minimum in the POC environment, before releasing the environment for general testing and evaluation.

| Step | Instructions |
|---|---|
| 1<br>☐ Failed<br>☐ Passed | **Pre-testing verification as a TFS Administrator**<br>• Logon to the application-tier (AT) server as an administrator using *TFSHOLPOC\TfsAdmin*.<br>Verify TFS Administration Console settings are showing correctly:<br>• Administration Console Users are limited to only those who should have access.<br>• Confirm the *DefaultCollection* Team Project Collection (TPC) is online and statuses are good. |
| 2<br>☐ Failed<br>☐ Passed | **Build Server pre-testing verification**<br>• Logon to the build server as an administrator using *TFSHOLPOC\TfsAdmin*.<br>Verify TFS Administration Console settings are showing correctly:<br>• Verify Build Server shows online from Visual Studio for the TPC.<br>• The drop folder has been created and shared with the appropriate users and groups. |
| 3<br>☐ Failed<br>☐ Passed | **Ability to create a new team project and assign permissions**<br>• Connect to TFS server from Visual Studio.<br>• Create a team project for your application based on the Scrum template.<br>• Set the project administrator and contributor accounts into the appropriate TFS group for the TFS project, using *TFSHOLPOC\TfsAdmin* and *TFSHOLPOC\TfsUser* respectively. |
| 4<br>☐ Failed<br>☐ Passed | **Ability to check-in code**<br>• Logon to the build server as *TFSHOLPOC\TfsUser*.<br>• Connect to the team project created in previous step.<br>• Create and check-in a test solution, and verify that no errors occur. |
| 5<br>☐ Failed<br>☐ Passed | **Create and run build**<br>• Logon to the build server as an administrator using *TFSHOLPOC\TfsAdmin*.<br>• Connect to TFS server from Visual Studio.<br>• Create a build definition for the test solution.<br>• Confirm the build runs successfully and the output is in the drop location. |
| 6<br>☐ Failed<br>☐ Passed | **Create a work item**<br>• Create a Product Backlog work item. You should be able to create the work item successfully. |
| 7<br>☐ Failed<br>☐ Passed | **Update a work item and create a linked work item from it**<br>• Update the Product Backlog work item you created and add a linked task.<br>• You should be able to create the task work item successfully. |
| 8<br>☐ Failed<br>☐ Passed | **Review Reporting Services site**<br>• Navigate to the Reports folder in Team Explorer. Choose and view a report.<br>• You should be able to access it. |

**Table 22 – POC Functional Testing**

# Additional Test Considerations

For more advanced TFS on Azure IaaS environment you can consider additional tests such as:

| Step | Instructions |
|---|---|
| 1<br>☐ Failed<br>☐ Passed | **Pre-testing verification as a TFS Administrator**<br>Verify TFS Administration Console settings are showing correctly:<br>• Administration Console Users are limited to only those who should have access.<br>• E-mail settings are configured.<br>• Updates are as expected.<br>• Create a new team project collection (TPC), i.e. *TPC01*.<br>• Confirm that the TPC is online and statuses are good. |
| 2<br>☐ Failed<br>☐ Passed | **TFS Health pre-testing verification**<br>• Best Practice Analyzer has been run and does not show any errors.<br>• If Reporting Services are available:<br>  ○ Install Grant Holliday's TFS Performance pack reports.[90] Verify that they run.<br>  ○ Install Grant Holliday's Administrative Report Pack reports [91]. Verify that they run.<br>  ○ Review Grant Holliday's What does a well maintained Team Foundation Server look like [92]<br>Make adjustments to align to his recommendations. |
| 3<br>☐ Failed<br>☐ Passed | **(If configured) Review SharePoint site**<br>• Stay in Visual Studio as the project administrator in the solution you were working in.<br>• Navigate to the SharePoint site for the team project. You should be able to access it. |
| 4<br>☐ Failed<br>☐ Passed | **(If configured) Review Reporting Services site**<br>• Stay in Visual Studio as the project administrator in the solution you were working in.<br>• Navigate to the Reports folder in Team Explorer. Choose and view a report. You should be able to access it. |
| 5<br>☐ Failed<br>☐ Passed | **(If configured) Review Cube Processing**<br>• Using Grant Holliday's Cube Processing report, navigate to the report and verify the cube processing is running. |
| 6<br>☐ Failed<br>☐ Passed | **Review TFS Internals Processing**<br>• Go to Event Logs and scan for any errors. |

**Table 23 – Recommended Functional Testing**

[90] http://blogs.msdn.com/b/granth/archive/2009/02/03/announcing-tfs-performance-report-pack.aspx

[91] http://blogs.msdn.com/b/granth/archive/2010/07/12/administrative-report-pack-for-team-foundation-server-2010.aspx

[92] http://blogs.msdn.com/b/granth/archive/2013/10/08/what-does-a-well-maintained-team-foundation-server-look-like.aspx

# In Conclusion

This concludes our adventure of considering, planning and deploying a TFS on Azure IAAS environment. We have touched on theory, introduced you to some of the important the planning decisions, and provided a proof-of-concept companion with the associated walkthroughs checklists.

We hope you find it a valuable technology to invest in and that you have found this guide useful.

Sincerely,

**The Microsoft Visual Studio ALM Rangers**