SIEMENS

Teamcenter Deployment Reference Architecture

Nov 28, 2022

Deployment Center 14.2

What's new in Deployment Reference Architecture for Teamcenter 14.2/Active Workspace 6.2/Deployment Center 14.2?

1. Best practices and guidelines in configuring and deployment of Store and Forward Volume

The "4.6 Distributed Global Deployment" section of this document is updated with best practices and guidelines in configuring and deploying Store and Forward Volume

What's new in Deployment Reference Architecture for Teamcenter 14.1/Active Workspace 6.1/Deployment Center 14.1?

2. Deployment enhancement supports Teamcenter deployment using shared TC_ROOT and TC_DATA

In the prior release of Teamcenter deployment, **Deployment Center (DC)** supported deployment of Teamcenter server using locally populated TC_DATA configuration directory.

In Deployment Center 14.1 and Teamcenter 14.1, DC allows to configure the Teamcenter Server deployment with shared TC_DATA configuration directory.

Quick Deploy configuration samples are also updated to use shared TC_DATA configuration directory.

3. Best practices and guidelines in configuring and maintaining TC_DATA configuration directory

The "Deployment Consideration" section of this document is updated with best practices and guidelines in configuring and maintaining TC_DATA configuration directory.

What's new in Deployment Reference Architecture for Teamcenter 13.2/ Teamcenter 14.0/Active Workspace 6.0/Deployment Center 4.2?

1. Microservice Deployment enhancements for Linux Platform to support Container Image Registry

In the prior release of Teamcenter and Active Workspace deployment, SPLM supported deployment of Microservice on the Docker Container on Linux platform using the service image deployed on local cache on each of the machine.

In the Deployment Center 4.2 and Active Workspace 6.0, SPLM enhanced to use a centralized container image registry that hosts all microservices that are installed during the deployment.

Quick Deploy configuration samples are also updated to use centralized container image registry that hosts all microservices.

2. Teamcenter Management Console deployment support

In the Deployment Center 4.2 and Teamcenter 13.3 and 14.0, Deployment Center supports deploying and maintenance of Teamcenter Management Console.

Quick Deploy configuration samples are updated to include Teamcenter Management Console now.

What's new in Deployment Reference Architecture for Teamcenter 13.2/Active Workspace 5.2/Deployment Center 4.1?

1. Mixed configuration of single sign-on on/off deployment support

In the prior release of Deployment Center/Teamcenter/Active Workspace, SPLM supported configuring and enabling of single sign-on (SSO) using Teamcenter Security Service on the entire Teamcenter environment, and there was no capability to turn off the single sign-on (SSO).

In the Deployment Center 4.1, SPLM enhanced Teamcenter Security Service and other components to support the turning on/off SSO at the individual component level and at the entire environment level. This capability is available via both Deployment Center UI and Quick Deploy Configurations.

This release also includes the enhanced Quick Deploy Configuration samples demonstrating turning on/off SSO at the individual component level based on business requirements.

2. High availability / loading balancing configuration for indexing engine using replica and ZooKeeper

In the Teamcenter Deployment Center Reference Architecture 4.1_v1, SPLM added high availability deployment configuration for the "Indexing Engine" using replica configuration and ZooKeeper orchestrator.

Deployment Reference Architecture diagram "4.4 Distributed Deployment with High Availability" is updated to demonstrate the replica configuration and ZooKeeper orchestrator.

3. Active Workspace Visualization Deployment support for Linux Platform

In the prior release of Teamcenter and Active Workspace deployment, SPLM supported deployment of Active Workspace Visualization on Windows platform only.

In the Deployment Center 4.1 and Active Workspace 5.2, SPLM enhanced Active Workspace Visualization to support deploying it on Linux platform as well.

Quick Deploy configuration samples for of Active Workspace Visualization on Linux are included now. You can use these configurations samples to deploy the Active Workspace Visualization on the Linux platform.

4. Deployment Reference Architecture diagram for Supplier Collaboration Foundation

In the Teamcenter Deployment Center Reference Architecture 4.1_v1, SPLM added Deployment Reference Architecture diagram for Supplier Collaboration Foundation.

It also includes Quick Deploy configuration samples for Supplier Collaboration Foundation installation for both Windows and Linux. You can use these configurations samples to deploy the Supplier Collaboration Foundation.

5. Forward Deployment for Active Workspace

Forward deployment of Active Workspace with Teamcenter Office Online and Visualization for the security regulations and performance reasons is added in section 4.6 Distributed Global Deployment. The corresponding changes to Quick Deploy configuration samples is included.

What was new in Deployment Reference Architecture for Teamcenter 13.1/Active Workspace 5.1/Deployment Center 4.0?

1. Deployment Reference Architecture Quick Deployment Configuration samples for Linux Platform

In the prior release of Teamcenter Deployment Center Reference Architecture, SPLM provided Quick Deployment configuration samples to install all the Teamcenter Deployment Reference Architecture on the Windows platform only.

In the Teamcenter Deployment Center Reference Architecture 4.1_v1, Quick Deploy configuration samples for Linux platform are included as well. Now you can use these configurations samples to deploy all the Deployment Reference Architectures on the Linux platform.

Quick Deploy Configuration examples for all the Teamcenter Deployment Reference Architecture are shipped as part of Teamcenter Deployment Reference Architecture download and can be located under the folder \quick_deploy_configurations\wntx64 in case of Windows platform or \quick_deploy_configurations\lnx64 for Linux platform.

2. Installation of multiple instances of Teamcenter Components and Configuration Assistance via Quick Deploy Configuration and Deployment Center UI

In the prior release of Deployment Center/Teamcenter/Active Workspace, SPLM supported installation of multiple of instances of many Teamcenter Components.

In the Deployment Center 4.0 and Teamcenter 13.1, SPLM added more Teamcenter Components that can be installed with multiple instances. Deployment Center is enhanced with configuration assistance that will guide you to configure the multiple instances and to cluster these components correctly. This capability is available via both Deployment Center UI and Quick Deploy Configurations.

This release also includes the enhanced Quick Deploy Configuration samples with the support of multiple Instances and clustering of Teamcenter Components in a single configuration file.

3. Unique/Subset set of Application configuration support of Teamcenter Rich Client

In the prior release of Deployment Center/Teamcenter, SPLM supported every installation of Teamcenter Rich Client (2 Tier or 4 Tier) with same set of Applications that are configured in the “3 Applications” tab.

In the Deployment Center 4.0 and Teamcenter 13.1, SPLM enhanced to install the unique/different set of Applications on each of the Teamcenter Rich Client (2 Tier or 4 Tier).

It also includes the enhanced Quick Deploy Configuration samples that provides the configuration syntax configure the different set of applications on Teamcenter Rich Client Component.

4. Unique/Subset set of translator configuration support of Dispatcher Module

In the prior release of Deployment Center/Teamcenter, SPLM supported every installation of Dispatcher Module component with same set of Translators that are configured in the “3 Applications” tab.

In the Deployment Center 4.0 and Teamcenter 13.1, SPLM enhanced to install the unique/different set of translators on each of the Dispatcher Module component.

It also includes the enhanced Quick Deploy Configuration samples that provides the configuration syntax configure the different set of translators on each of the Dispatcher Component.

5. Alias/Alternate name configuration Installation support with Teamcenter Connection Override

In the prior release of Deployment Center/Teamcenter, SPLM supported configuration of Teamcenter deployment with physical host, and Deployment Center automatically connected to machine using physical host name only.

In the Deployment Center 4.1 and Teamcenter 13.1, SPLM enhanced to configure Teamcenter components using alias name or alternate name, and by default Deployment Center automatically connects/wires the component using specified alias/alternate names. Now you can override this default configuration according to your IT security standards.

It includes the enhanced Quick Deploy Configuration samples that provides the configuration components using alias and overrides default configuration with another alternate name.

1 Introduction

A strong understanding of Teamcenter's basic deployment architecture is essential to efficiently deploy Teamcenter and Active Workspace using Deployment Center or the TEM installer.

The following groups can use this reference of basic deployment architectures to pre-plan and prepare their hardware/software infrastructure with the appropriate network security.

- Business owners/sponsors
- Teamcenter administrators
- Database administrators
- Enterprise architects
- Project managers
- Project teams
- Build teams

This reference lists recommended deployment architectures that are tested and validated by Siemens. It also provides details about how to use basic architecture components to correctly deploy Teamcenter Foundation and Active Workspace, including:

- Which tiers you should place specific components in.
- Communication flow between components and the parameters used for the communication.
- Component dependency and the related sequence of deployment.

This document covers the deployment specific to Teamcenter 12.4/13/13.1/13.2/13.3/14.0/14.1/14.2 and Active Workspace 5.0/5.1/5.2/6.0/6.1/6.2 using Deployment Center 3.2/4.0/4.1/14.1/14.2. However, these deployment architectures can be used as a general reference to other releases. The recommended deployment architectures documented here are tested and validated by Siemens.

2 Deployment Reference Architecture

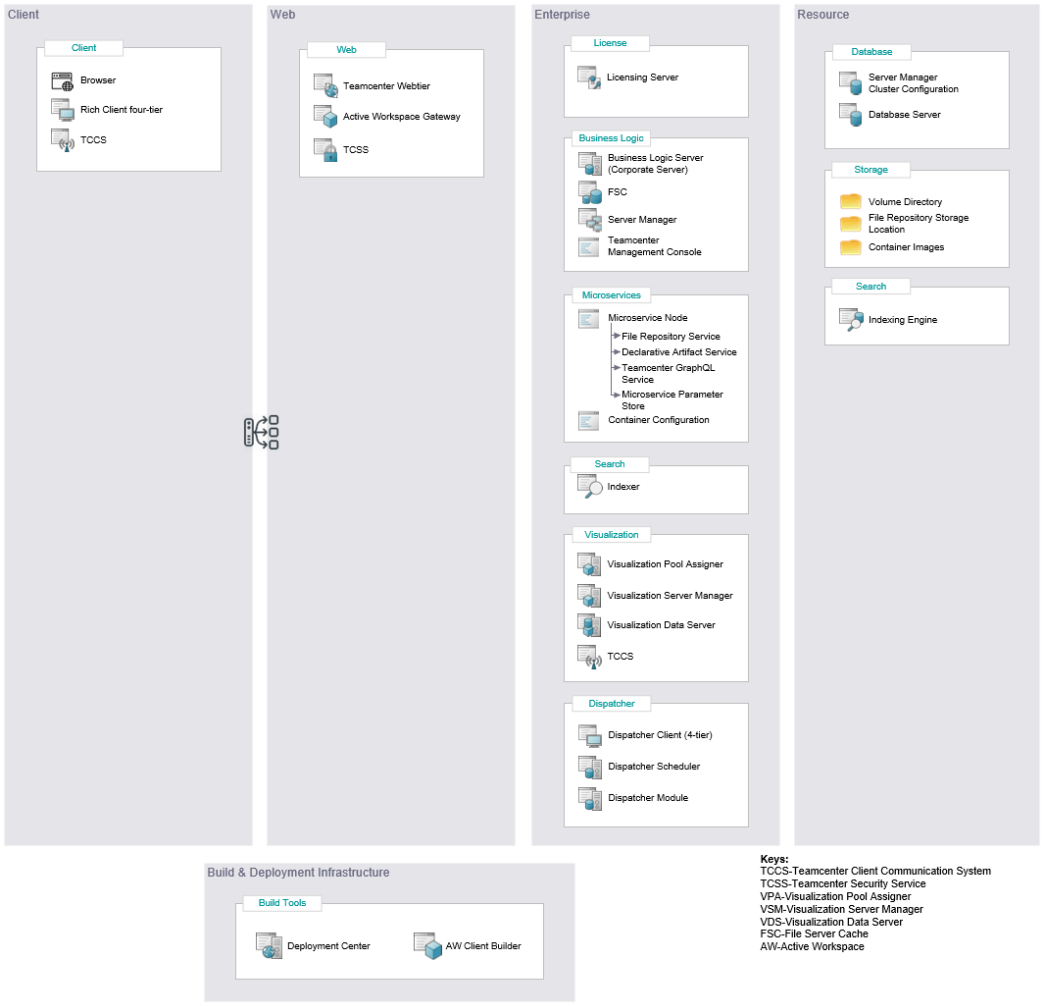
The following diagrams show all the basic deployment architecture components involved in deploying Teamcenter and Active Workspace.

2.1 Deployment Architecture Components and Tiers

The following diagram shows the basic components of Teamcenter and Active Workspace placed in the appropriate tiers – it's recommended that you plan your components in these recommended tiers. The boxes indicate their logical groupings and the recommended best practices for deploying specific components together.

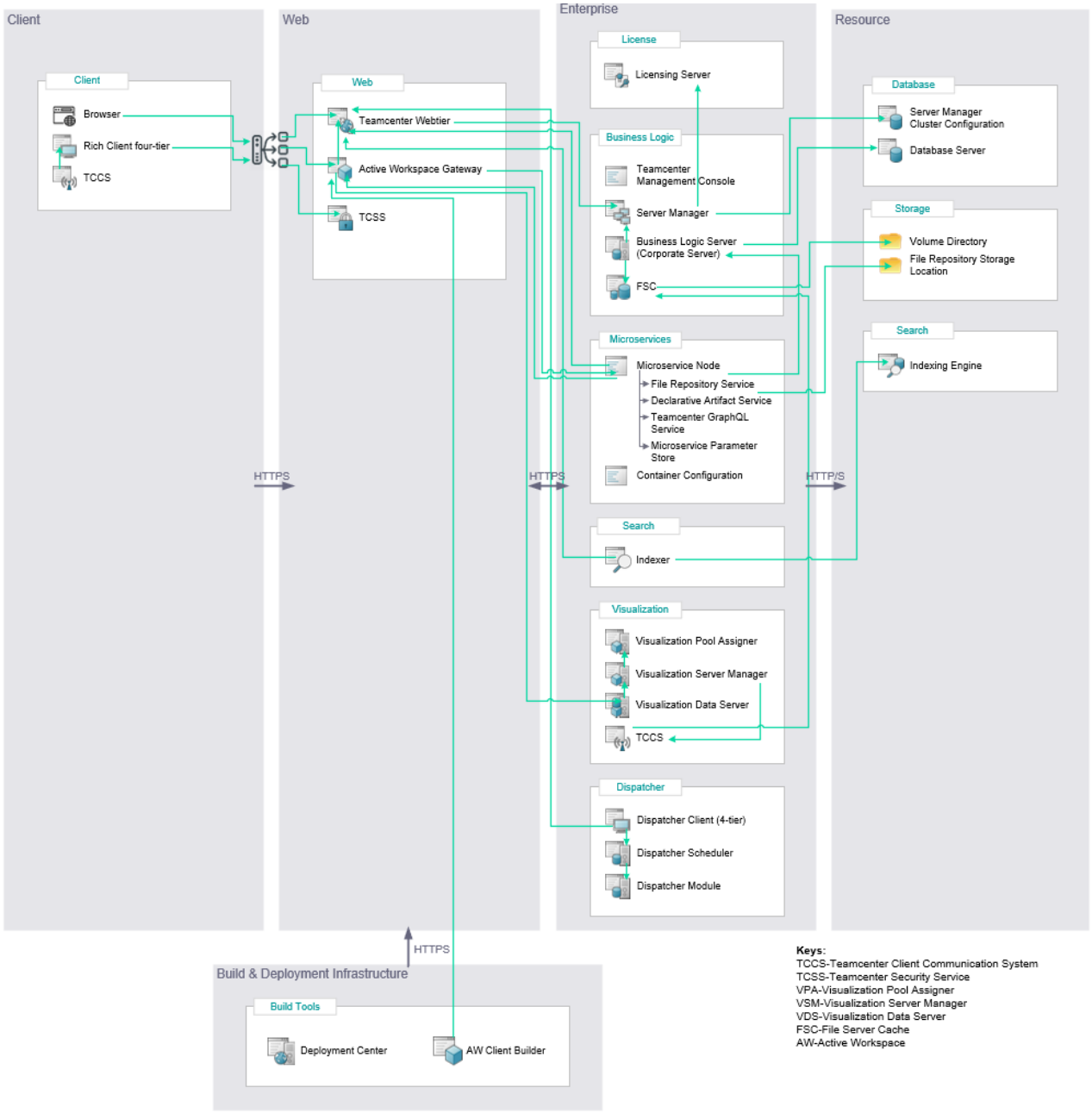
The following specific components were added at Teamcenter 12.3 and Active Workspace 4.3 onwards. Adhere to the recommended best practices for these components for successful deployments.

- Microservice Node and Microservices
- Active Workspace Gateway & Active Workspace Client Builder
- Visualization Pool Assigner ,Visualization Server Manager



2.2 Connection and Communication Details (at configuration time)

The following diagram shows the communication between components and the direction of the communication flow. Use this information to adjust your network security to allow secure communication between components.



Refer to the **\document\Teamcenter_Deployment_Connection_and_Communication_Table.xlsx** Excel spreadsheet for additional details about each of the connections illustrated in the previous graphic. It is shipped with the Teamcenter Deployment Reference Architecture downloads. The spreadsheet lists:

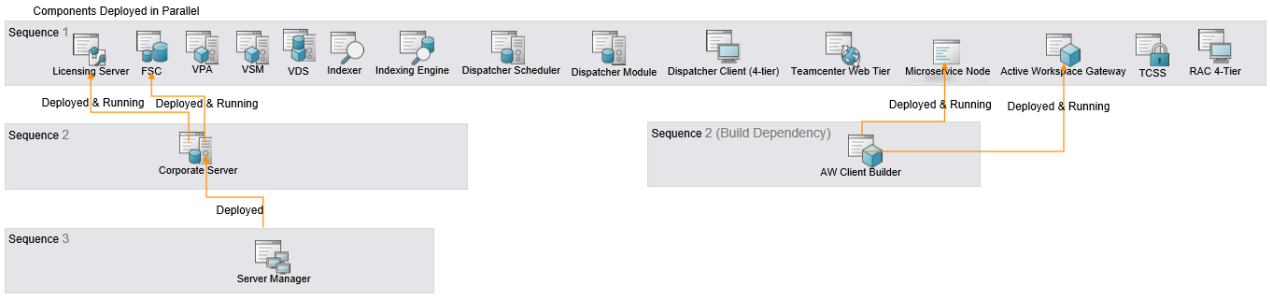
- The property used in creating the connection

- The component from which the connection must be configured
- The protocols used for the connection
- Any default port values
- Latency/network performance

Use this information to configure the components correctly, and to identify where you need to adjust network securities to ensure communication through the ports. You may use this spreadsheet as a starting template and update according to your environment that you plan to deploy.

2.3 Deployment Sequence (at deployment time)

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence. The following diagram shows the sequence dependency during deployment and/or during runtime. The components listed as Sequence 1 can be deployed in any order, followed by any components in Sequence 2, and then followed by component in Sequence 3.



3 Deployment Considerations

Build a secure and reliable Teamcenter environment, with optimized cost, by applying the following considerations.

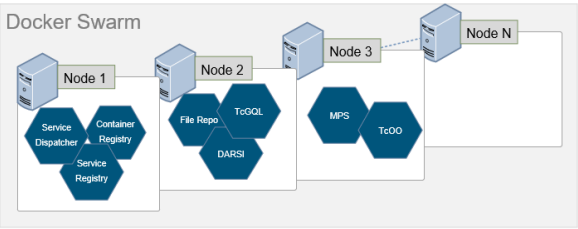
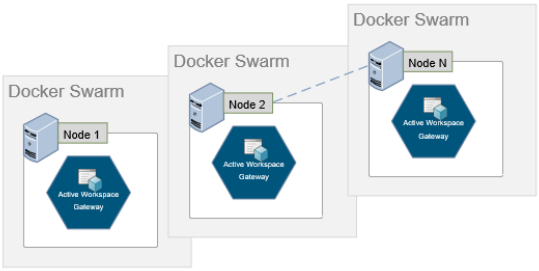
- Operating systems
- Hardware: Physical versus virtual
- Security
- High availability
- Scalability
- Environment types
- Global environments
- Backup and restore
- Optional software components

3.1 Operating Systems

You can choose Linux or Windows as the operating system for your server infrastructure and your clients. Teamcenter offers deployment solution on both platforms. The deployment options are similar, with the following exceptions:

Microservice and Active Workspace Gateway configuration differs between Linux and Windows as follows:

	Linux 64-bit	**Windows 64-bit**
Third party software required	Docker and a container registry must be installed on the Linux host before you can install the microservice framework.	None
Microservice Node/Active Workspace Gateway Deployment	On Linux Teamcenter microservices/Active Workspace Gateway run as containers in a Docker swarm. A container runtime Docker is a pre-requisite to install Teamcenter microservices/Active Workspace Gateway in a Linux environment. From Active Workspace version 6.0 onwards Teamcenter microservices and Active Workspace Gateway container images are deployed into the container registry, hence a container registry is also a pre-requisite. For microservices installation details on Linux, refer to the “Installing Teamcenter microservices” section from the Active Workspace installation document. Microservices run as containers in Docker Swarm:	On Windows Teamcenter microservices/Active Workspace Gateway run as processes. On all machines where a microservice node is installed, a Windows service called Teamcenter Process Manager will implicitly get installed. This service will start/stop microservices/Active Workspace Gateway. Whenever the Teamcenter Process Manager is stopped/restarted, all the microservices/Active Workspace Gateway running on the same

	 Active Workspace Gateway run as containers in Docker Swarm: 	machine will be stopped/restarted as well. For microservices installation details, refer to the “Installing Teamcenter microservices” section from the Active Workspace installation document.

NOTE: Docker on Linux requires that IP forwarding is enabled on the host machine for successful communication between Docker containers and the host machine. Refer to **SFB-TEAMCENTER-8171965**.

3.2 Hardware – Physical versus Virtual

Teamcenter can be configured either on physical machines or on VMs. With industries trending toward virtualization, you can take advantage of the benefits that virtual machines offer in terms of cost, physical footprint, efficiency, disaster recovery and high availability.

Hardware considerations include CPU, memory, network I/O, and disk I/O. Your hardware requirements will vary based on the following deployment considerations:

- Security: Multi-tier deployment with client authentication
  - Web Tier: This generally requires a low-end server box with very high I/O and lower CPU.
  - Enterprise Tier: This mostly requires a server box with lots of memory and CPU capacity, network isolated.
  - Resource Tier: This mostly requires a server box with lots of memory and CPU capacity, network isolated, and mass storage capacity.
- High availability environments: Varies based on your failover approach. For example, replicating the entire system versus replicating critical components versus warm/cold standby.
- Scalability: Hardware requirements and number of machines varies based on sizing and the number of concurrent users. Also based on dynamic/calculated scaling configurations.
- Global: Varies based on the number of geographical regions, the types of functionality/service configured to provide in the specific regions, as well as all the previous considerations.

- Environment types: Varies based on the type of environment and the complexity of the configuration. Different environments (development, testing, production) require different deployment strategies, which produce different hardware requirements.
- Optional software components: Additional hardware or tuning may be required based on any optional software. Depending on the optional software's hardware requirements, you can choose to configure it on an existing Teamcenter server or onto a dedicated server.

3.3 Security

You can deploy Teamcenter on either multi-tier or multi-layered architecture. Both deployments offer secured communication protocols that allow you to protect your data and intellectual property.

Use the following security options to deploy your Teamcenter environments according to your company's business processes and security protocols.

- Multi-tier with secured communication protocol configuration
- Client authentication challenging using Single Sign-on (Teamcenter Security Service) configuration
- Client authentication challenging using forward/reverse proxy (Teamcenter Client Communication System) configuration.

3.4 High Availability

The need for a high availability environment is based on the criticality of the business operation and the time it takes to recover from failure. If the business operation demands a permanent standby environment, then we highly recommend configuring such an environment, though doing so incurs additional costs for hardware and infrastructure maintenance.

You can configure high availability environments using the following strategies:

- One large resource configured as the primary setup for the environment and a second, smaller resource configured as a standby warm sever that always runs. Connect the second resource through a network switch to serve when the primary setup fails.
- One large resource configured as the primary setup for the environment and a second, smaller resource configured as a standby cold sever that is powered off. Connect the second resource through a network switch to power on and serve when the primary setup fails.
- Use multiple small resources to reduce the impact of a single failure on the overall system. Distribute requests across multiple resources using a load balancer configuration.
- If you configured the environment with redundant components to meet scalability requirements, the same redundant components can be configured to provide a failover setup.

3.5 Scalability

You can change the deployment configuration of an existing Teamcenter environment to scale up and down, based on the number of concurrent users to be supported.

Scaling an environment involves the following considerations:

- The number of concurrent users, and the limitation of hardware size for the demand.
- The components that should be considered for scaling, and the number of instances required to be configured.
- What components should be clustered to optimize the infrastructure and maintenance cost.

Teamcenter provides deployment solutions for the following scaling configuration strategies. See Section 4.4 for details.

- Configuring the Teamcenter component for scaling to utilize the hardware resource at its full capacity.
- Configuring software load balancer to balance the requests load efficiently across peer components.

Configuring hardware load balancer to efficiently distribute the load across the hardware resource to provide the real-time response to users.

3.6 Global Environments

Global business operations require additional deployment considerations and infrastructure maintenance for their extended Teamcenter environments to be successful. Global operations will also need to account for faster data exchange, data protection across networks, and international exchange compliances.

You can lower global deployment costs by configuring a centralized data infrastructure in a single region and distributing Client and Web tiers with file management system across your other geographical regions.

3.7 Environments – Development versus Testing versus Production

Before moving into a production environment, environment changes are created in a development environment, then reviewed and tested in testing environments. Your deployment strategies will vary, based on the type of environment.

- Development: Mostly single machine deployment, with development tools for developers.
- SIT (System Integration Testing): Mostly two-tier distributed setups, with fewer machines than in production.
- UAT (User Acceptance Testing)/Sandbox: Distributed setup with a secured multi-tier configuration, with fewer server machines than in production.
- Production: Distributed setup with a secured multi-tier configuration that is scalable and has high availability. Multiple high-end server machines are required.

3.8 Backup and Restore

Whether it is a hardware failure or it's someone accidentally deleting an important file, maintaining frequent backups is extremely important and can save countless hours of unnecessary work.

Teamcenter provides best practices, deployment considerations and instructions to back up and restore your environment when failure occurs. See Section 4.6 for details.

3.9 Optional Software Components

Teamcenter provides many optional capabilities for you to choose, based on your company's functional requirements. Additional hardware or tuning may be required based on any optional software. Depending on the optional software's hardware requirements, you can choose to configure it on an existing Teamcenter server or onto a dedicated server.

3.10 Configuring and maintaining Teamcenter Data (TC_DATA)

The Teamcenter data/TC_DATA is configuration directory that is consumed by the Teamcenter Server/utilities at runtime. This directory gets populated during deployment and is consumed during data model deployment. The Deployment Center provides an option to configure the Teamcenter Server with locally populated with TC_DATA configuration directory or a shared directory which is already populated. Apart from deployment option, any incorrect modifications to the directory or content after the initial deployment could result into Teamcenter server/utilities fatal failures at runtime. The following recommended best practices and guidelines must be followed while configuring and maintaining Teamcenter Data/TC_DATA configuration directory.

- Highly recommended to configure Teamcenter Server deployment to use locally populated TC_DATA configuration directory to maximize the Teamcenter performance.

  The Deployment Center is designed to effectively handle the configuration & maintenance of TC_DATA directory for the additional Tc Server deployment. Deployment Center automatically synchronizes all additional Tc Server TC_DATA directories when additional applications are deployed or updated/patched to latest Tc version. By default, the Deployment Center configures local TC_DATA directory and populates it on the machine where Teamcenter Server is configured to deploy.

  Configuring with local TC_DATA directory if business helps in
  - Better Teamcenter performance.
  - Better Indexing performance where Indexer deployment is configured with dedicated Teamcenter Server.

Ensure to synch all Tc Server locally populated TC_DATA directories with any manual updates made to Corporate Server TC_DATA directory.

- You may configure Teamcenter Server deployment to use shared TC_DATA configuration directory to minimize the cost in mass deployment of Teamcenter Server but beware of lower performance implications.

  As mentioned above SPLM highly recommends local TC_DATA configuration directory, however if business requirements demand to use shared TC_DATA directory then the Deployment Center provides option to specify the location of mounted shared TC_DATA directory for the Tc Server deployment.

  Configuring to use mounted shared TC_DATA directory has some advantages.
  - Reduces cost in deploying/synchronizing TC_DATA directory for the large number of additional Teamcenter Server.
  - Reduces cost in manually updating TC_DATA directory with any customer specific configurations on large number of additional Tc Servers.
  - Reduces cost in deploying/synchronizing TC_DATA directory of the Tc Server for large number of Teamcenter Rich Client 2-Tier.
  - Reduces cost in deploying/synchronizing TC_DATA directory of the Tc Server for large number of Teamcenter Rich Client 2-Tier configured connect to multiple Tc database/environment.

  To prevent deployment failures, ensure to turn off Antivirus/Windows Defender scanning during deployment or exclude the mapped TC_DATA drive from scanning.

  To prevent Teamcenter Server/utilities failures, ensure to configure Teamcenter Server deployment to use shared TC_DATA directory of the same OS platform as Teamcenter Server.

  Note: Configuring Teamcenter Server with shared TC_DATA directory may experience Teamcenter performance degradation depending on the hardware specification/network latency.

- Do not move/rename/delete TC_DATA configuration directory after the deployment to prevent Teamcenter Server/utilities failures.

  SPLM highly recommends to not to move/rename/delete TC_DATA after it is created as there are several Teamcenter component/deployment configurations are referenced to TC_DATA directory and changing location of the directory will result functionality failure of the referencing component.

  If business requirements demand to move/rename TC_DATA directory, then follow the instructions as below.

- Take a full backup of environment.
- Bring Teamcenter environment down.
- Backup Database.
- Perform a new installation of Teamcenter matching your environment on a different/same server with the desired/correct TC_ROOT/TC_DATA paths.
- Restore the Database backup to the new server.
- Test the new environment.

- Set correct TC_DATA directory for testing patching/upgrade to prevent Deployment failures

  Do not setup test environment manually, it may corrupt the environment during test upgrade/patching if TC_DATA directory points to incorrect/production TC_DATA location.

  As the newer version of Deployment Center is enhanced to support cloning of Teamcenter environment, SPLM recommends using this cloning functionality to clone an environment. It guides you to set up TC_DATA directory correctly for test environment.

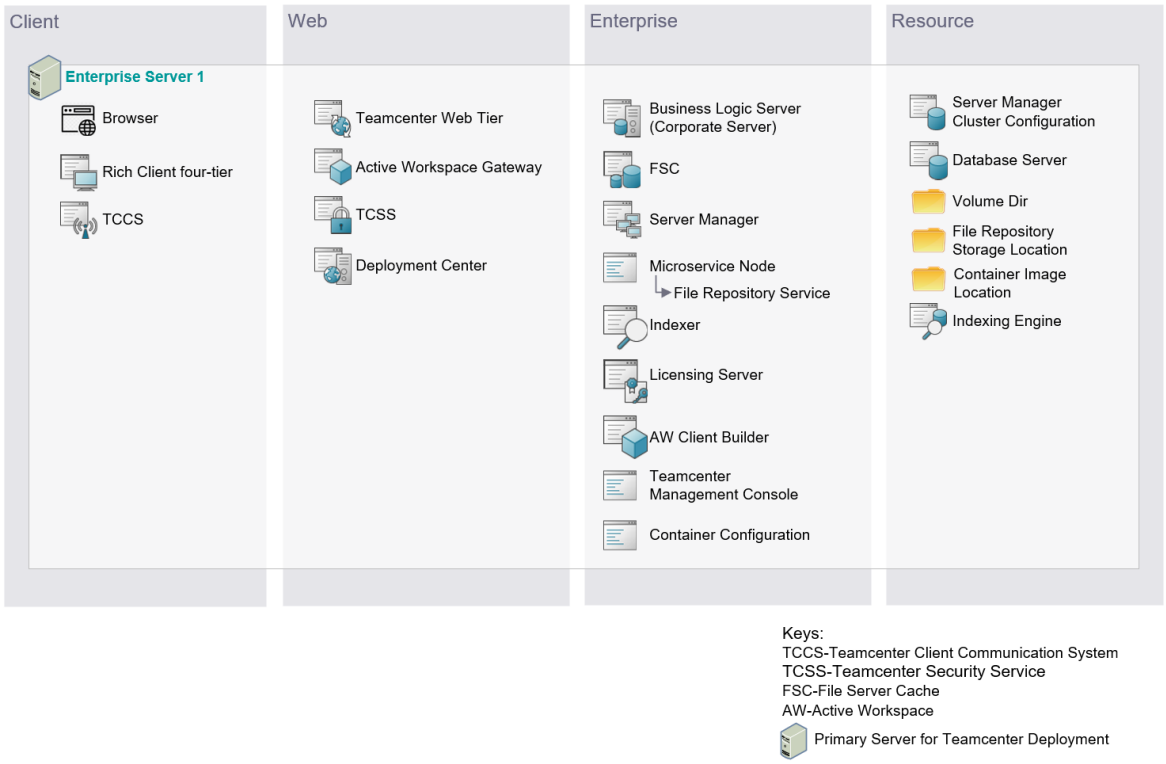
4 Deployment Reference Architecture Variations

The following sections document best-practice deployment architectures for different scenarios. Compare and match your deployment needs to one of these or a combination of these deployment architectures and follow that to plan your deployment.

4.1 Minimal One Box Deployment

Use this deployment for testing or development environments, and for training and demonstrations. The following diagram shows all the required deployment architecture components for this minimal deployment. They are all configured to deploy on single server machine.

As a variation of the single server deployment, one may use a Licenser Server on a dedicated or common server. The License Server could support multiple environments such as development, sandbox, testing, and production.



Configuration Instructions

This section briefly provides instructions to configure and deploy this “Minimal One Box Deployment” configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is shipped as part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.

     \quick_deploy_configurations\wntx64\Teamcenter_RA4.1_Minimal_Onebox_Deployment_wntx64.xml" in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.1_Minimal_Onebox_Deployment_lnx64.xml" in case of Linux platform.

   - and follow the instruction given in the readme "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

2. **Configure and deploy interactively using Deployment Center Client**

   A single server deployment is Deployment Center's default configuration. On selection of Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration and automatically selects the basic Teamcenter/Active Workspace application and all the required components.

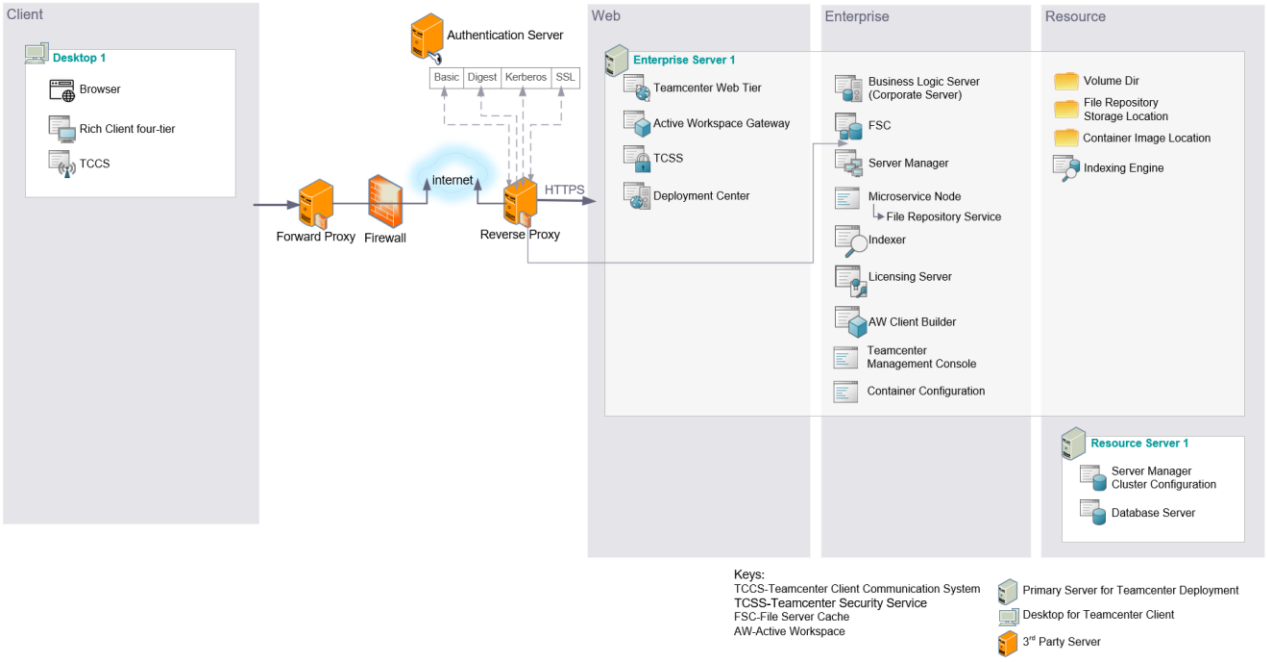
   When you assign a machine to one of the components, Deployment Center automatically configures the same machine to all the components except the Database Server and Licensing Server for which you have the option to assign the same machine, or a different machine.

   The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

4.2 All in One Box with secured communication protocol (HTTPS) Deployment

Use this deployment for production and for UAT environments. The following diagram shows all the required deployment architecture components for this one server deployment. They are all configured to deploy on single server machine.

A variation of the single server deployment would use Database on the same server and a Licenser Server on a dedicated or common server. The Database Server/License Server could support multiple environments such as development, sandbox, testing, and production.



Configuration Instructions

This section briefly provides instructions to configure and deploy this "All in One Box with secured communication protocol (HTTPS) Deployment" configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration. \quick_deploy_configurations\wntx64\Teamcenter_RA4.2_All_in_Onebox_Secured_Deployment_wntx64.xml" in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.2_All_in_Onebox_Secured_Deployment_lnx64.xml" in case of Linux platform.

   - and follow the instruction given in the readme "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

2. **Configure and deploy interactively using Deployment Center Client**
    - A single server deployment is Deployment Center's default configuration. On selection of Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration and automatically selects the basic Teamcenter/Active Workspace application and all the required components.
    - When you assign a machine to one of the components, Deployment Center automatically configures the same machine to all the components except the Database Server and Licensing Server for which you have the option to assign the same machine, or a different machine. For this configuration, assign same machine to both Database Server and Licensing Server components.
    - To assign the Client Desk machine to RAC component, change the default **Environment Type** from **Single Box** to **Distributed** on the **2 Options** tab and assign the Desk machine to RAC component.
    - Follow the instructions given in the Appendix Section 5.22 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.
    - Follow the instructions given in the Appendix Section **Error! Reference source not found.** to e nable the Teamcenter Security Service behind a firewall and TCCS on the Client tier to communicate using forward proxy on the client side, and reverse proxy on the server side.
    - The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

Configuration Matrix

Following configuration matrix table shows a few of the configuration variations available to choose while configuring Teamcenter/Active Workspace deployment components via Deployment Center client or Quick Deploy configuration xml based on business needs.

Conditions	**1**	**2**	**3**	**4**	**5**
Operating System	Windows	Windows	Windows	Linux	Linux
Database Server	MsSQL	Oracle	Oracle	Oracle	Oracle
Security Service	No	Yes	Yes	Yes	No
HTTPS	Yes	Yes	Yes	Yes	Yes
Forward Proxy	Yes	No	Yes	No	Yes
Reverse Proxy	None	Kerberos	PKI	Basic/Digest	None
Web tier Type	.Net	JavaEE	JavaEE	JavaEE	JavaEE
WebApp Server Type	NA	Jboss	Websphere	Tomcat	Weblogic

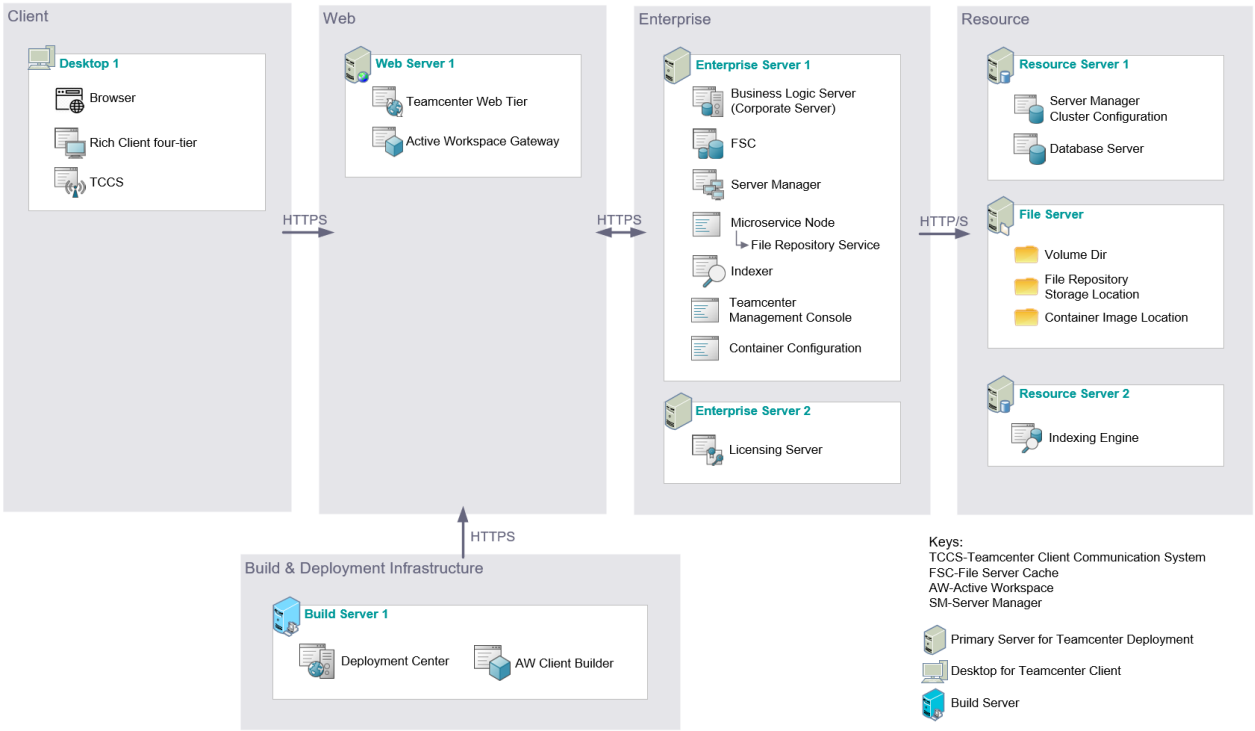
4.3 Distributed Deployment with Security

Teamcenter can be configured to deploy on a multi-tier environment where the Client, Web, Enterprise, and Resource tiers are configured with different security protocols, based on your company's security requirements.

4.3.1 Multi-tier with Secured Communication Protocol (HTTPS)

It is highly recommended to configure your environment to use secured communication protocols between the components across the tiers.

An optimal deployment requires placing the components in the appropriate tier. The diagram in Section 2.1 illustrates which tiers to place components in, and Section 2.2 provides communication protocol information that can be used during deployment configuration. The following diagram shows an example of a four-tier deployment configuration with HTTPS.



A variation of the multi-tier deployment would use a Licenser Server on a dedicated or common server. The License Server could support multiple environments such as development, sandbox, testing, and production.

Configuration Instructions

This section briefly provides instructions to configure and deploy this "Multi-tier with Secured Communication Protocol (HTTPS)" configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.

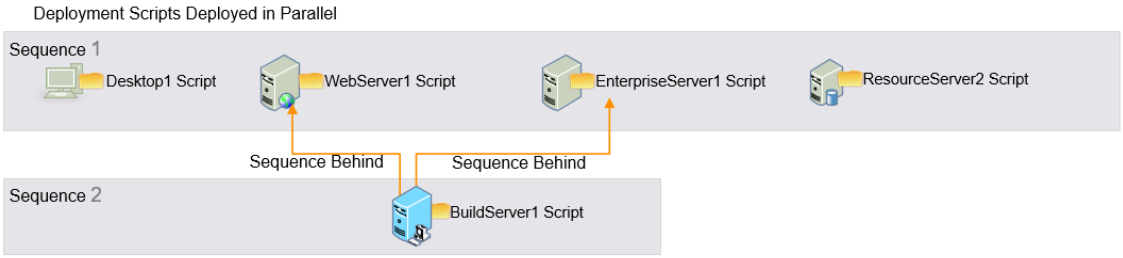
     \quick_deploy_configurations\wntx64\Teamcenter_RA4.3.1_Multitier_HTTPS_Deployment_wntx64.xml" in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.3.1_Multitier_HTTPS_Deployment_lnx64.xml" in case of Linux platform.

   - and follow the instruction given in the readme "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

2. **Configure and deploy interactively using Deployment Center Client**
   Deployment Center makes configuring a multi-tier deployment straightforward by offering the option to choose a distributed deployment. Use the following general instructions to configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.
   - When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
   - On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
   - It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
   - We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
   - To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
   - The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

- Desktop1 Script
- WebServer1 Script
- EnterpriseServer1 Script
- ResourceServer2 Script

And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.

- BuildServer1 Script must be run after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.

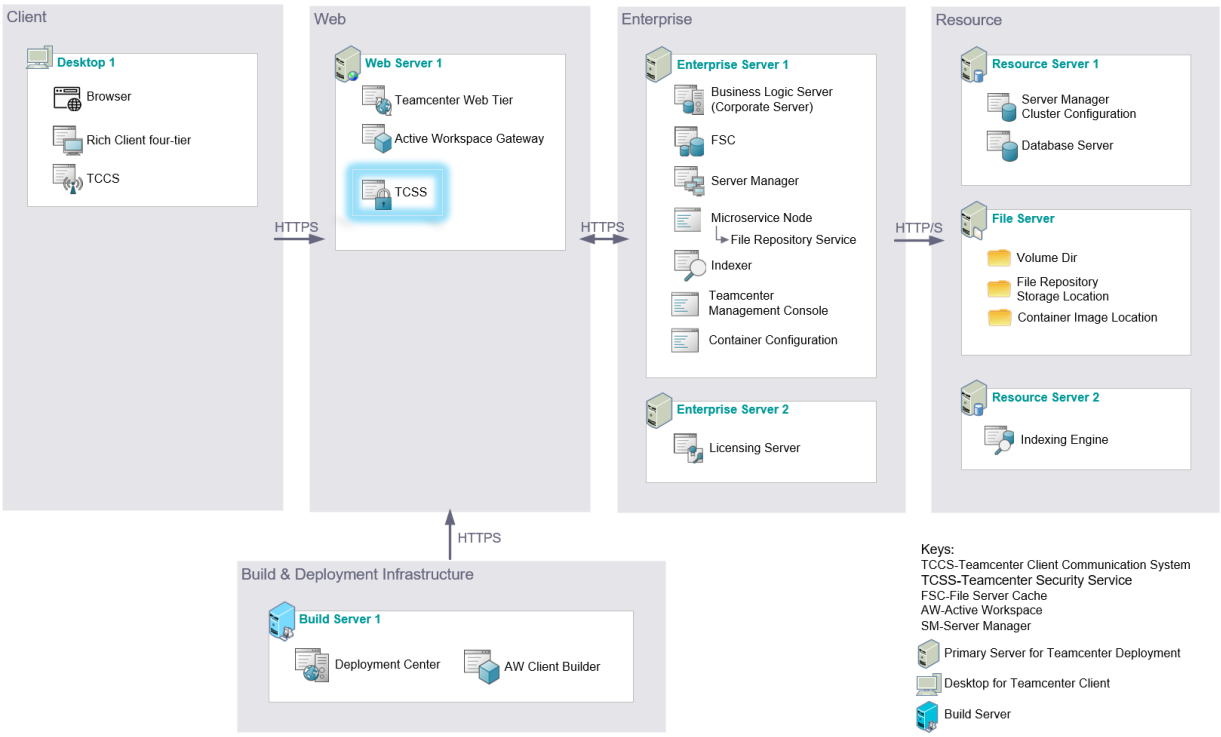
4.3.2 Multi-tier with Client Authentication using Single Sign-on (Teamcenter Security Service)

The default Teamcenter deployment configuration offers client authentication challenges based on the company's organization configured within Teamcenter by organizing user accounts and their respective permissions and user groups. Alternatively, you can configure increased authentication challenges using Teamcenter Security Service (TCSS). This strategy allows you to configure the LDAP Directory Server to authenticate Teamcenter user sets in the directory server and grants application-level authorization.

The following diagram shows another variation of multi-tier security configuration with a Teamcenter Security Service (TCSS) component on the web server to enable Single Sign-On application-level authentication using a Login Service and Identity Service configuration.

NOTE: The following deployment does not configure the Security Services behind a firewall. You must deploy the Identity Service on a web server using SSL with server certificates.

A variation of this deployment would use a Licenser Server on a dedicated or common server. The License Server could support multiple environments such as development, sandbox, testing, and production.



Configuration Instructions

This section briefly provides instructions to configure and deploy this "Multi-tier with Client Authentication using Single Sign-on (Teamcenter Security Service)" configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**

- Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.

  \quick_deploy_configurations\wntx64\Teamcenter_RA4.3.2_Multitier_SingleSignOn_Deployment_wntx64.xml" in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.3.2_Multitier_SingleSignOn_Deployment_lnx64.xml" in case of Linux platform.

- and follow the instruction given in the readme "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

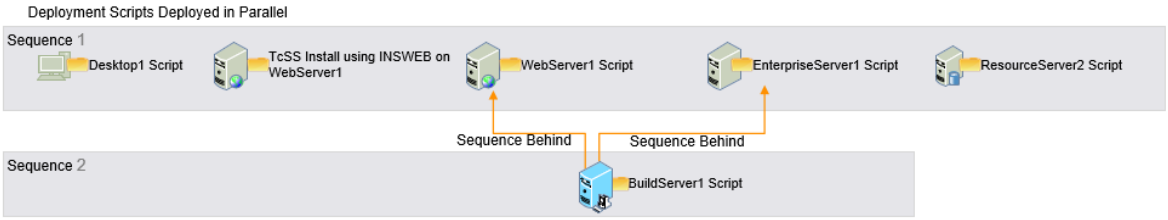
2. Configure and deploy interactively using Deployment Center Client

Deployment Center makes configuring a multi-tier deployment straightforward by offering the option to choose a distributed deployment. Use the following general instructions to configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.

- When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
- On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
- It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
- We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
- To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
- Follow the instructions given in the Appendix Section 5.22 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.
- The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

- Desktop1 Script
- WebServer1 Script
- EnterpriseServer1 Script
- ResourceServer2 Script

And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.

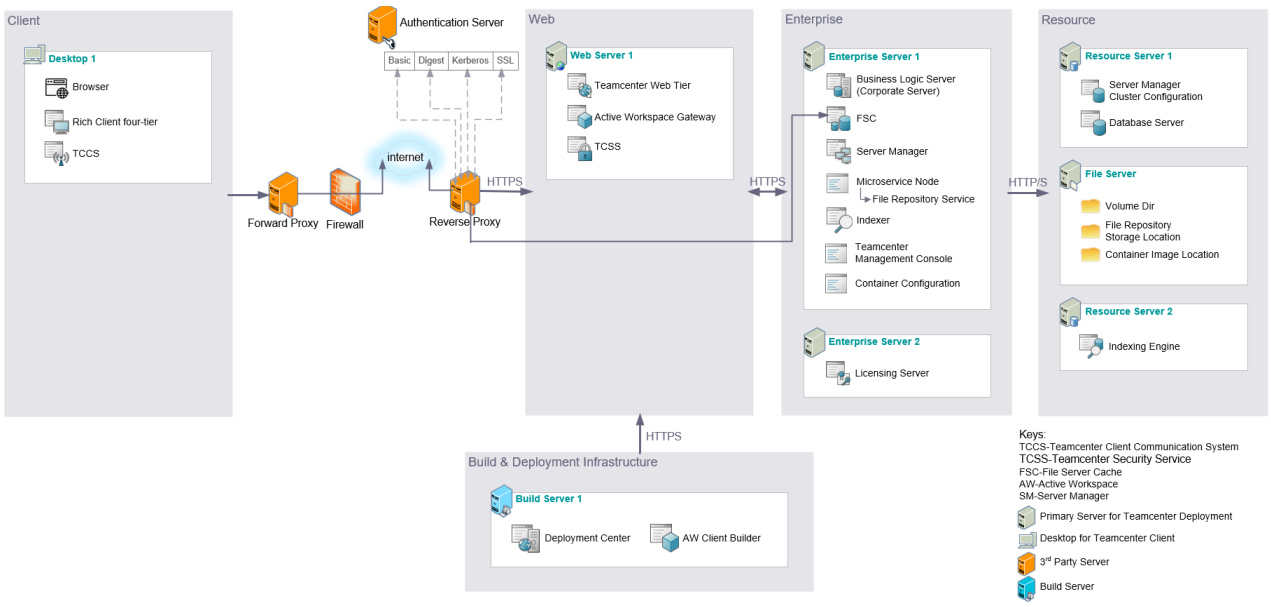
- BuildServer1 Script must be run after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.

4.3.3 Multi-tier with Client Authentication using Forward/Reverse Proxy (Teamcenter Client Communication System)

In addition to the Client authentication challenging using Single Sign-on (Teamcenter Security Service), Teamcenter offers deployment solution to configure Security Services (Login Service and Identity Service) deployed behind a firewall where no SSL is needed on web server using Server certificates. Clients do not see Security Service URL traffic, instead use client configured with Forward/Reverse Proxy using Teamcenter Client Communication System (TCCS) and the Forward/Reverse Proxy server firewall that guarantees communication between them is secure.

The following diagram shows the deployment configuration to deploy Security Services (Login Service and Identity Service) behind a firewall. The Teamcenter Client Communication System on client is enabled to communicate through Forward Proxy on client side and Reverse Proxy on server side with the Basic or Digest or Kerberos or SSL or no authentication options.

The Forward Proxy/Reverse Proxy and authentication types are configured using a third-party solution and these configuration details are specified during the deployment configuration of Teamcenter Client Communication component on the Client.



A variation of the above configuration could be to have Licenser Server on the dedicated/common server that would serve the multiple types of environments such as development, demo/training, sandbox, testing and production.

Configuration Instructions

This section briefly provides instructions to configure and deploy this “Multi-tier with Client Authentication using Forward/Reverse Proxy (Teamcenter Client Communication System)” configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.

     \quick_deploy_configurations\wntx64\Teamcenter_RA4.3.3_Multitier_ForwardReverseProxy_Deployment_wntx64.xml" in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.3.3_Multitier_ForwardReverseProxy_Deployment_lnx64.xml" in case of Linux platform.

   - and follow the instruction given in the readme "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

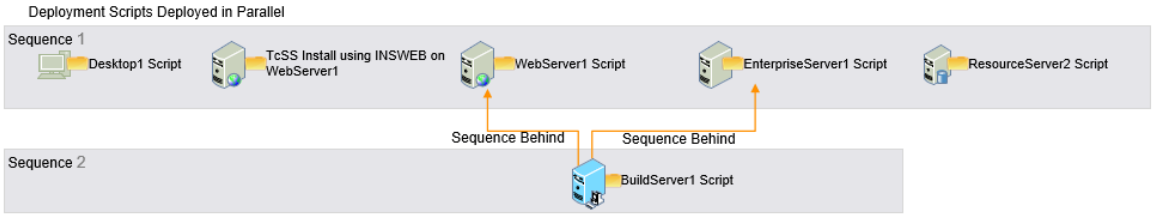
2. **Configure and deploy interactively using Deployment Center Client**

   Deployment Center makes configuring a multi-tier deployment straightforward by offering the option to choose a distributed deployment. Use the following general instructions to configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.

   - When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
   - On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
   - It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
   - We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
   - To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
   - Follow the instructions given in the Appendix Section 5.2 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.
   - Follow the instructions given in the Appendix Section **Error! Reference source not found.** to e nable the Teamcenter Security Service behind a firewall and TCCS on the Client tier to communicate using forward proxy on the client side, and reverse proxy on the server side.
   - The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

- Desktop1 Script
- WebServer1 Script
- EnterpriseServer1 Script
- ResourceServer2 Script

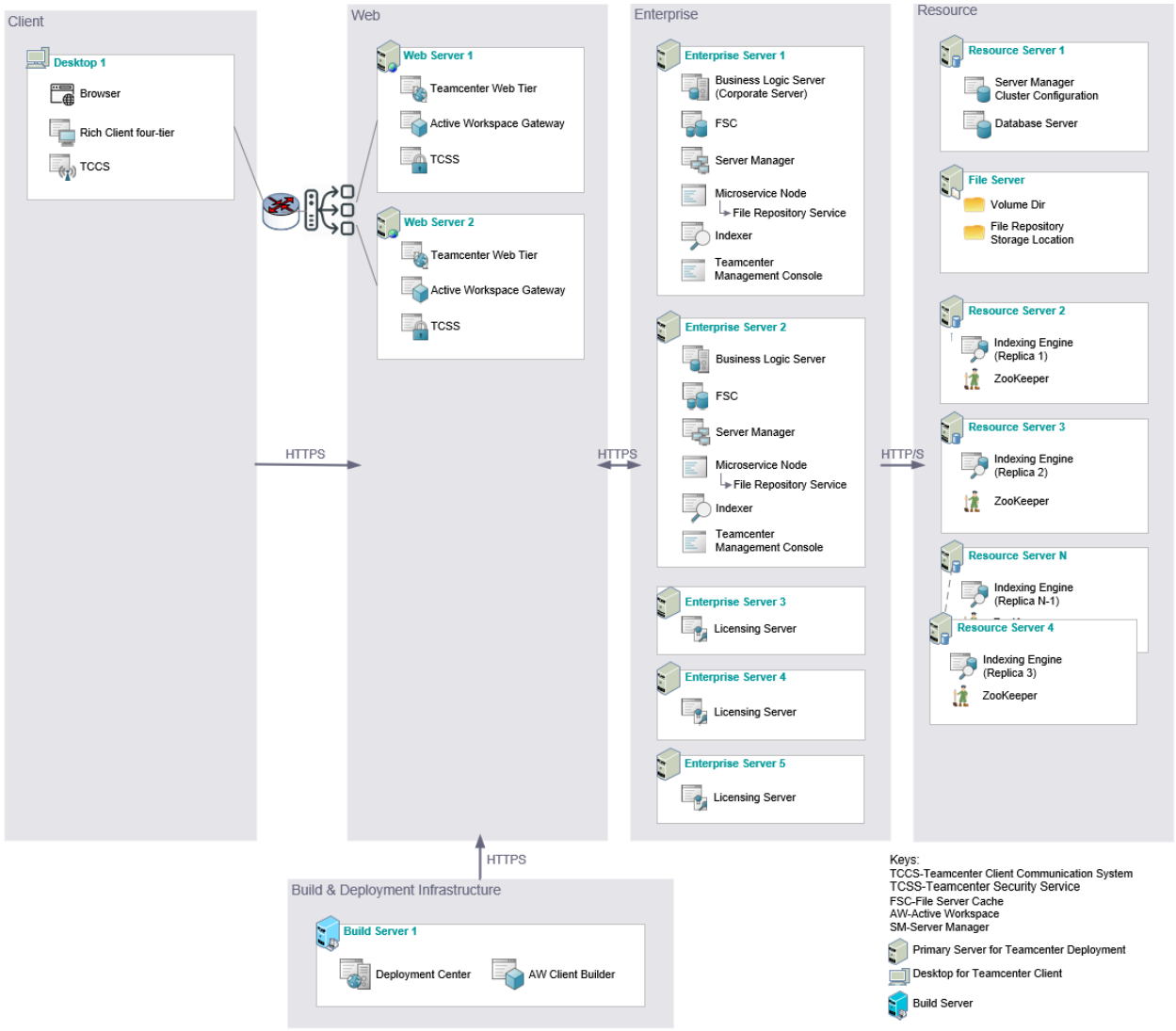
And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.

- BuildServer1 Script must be run after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.

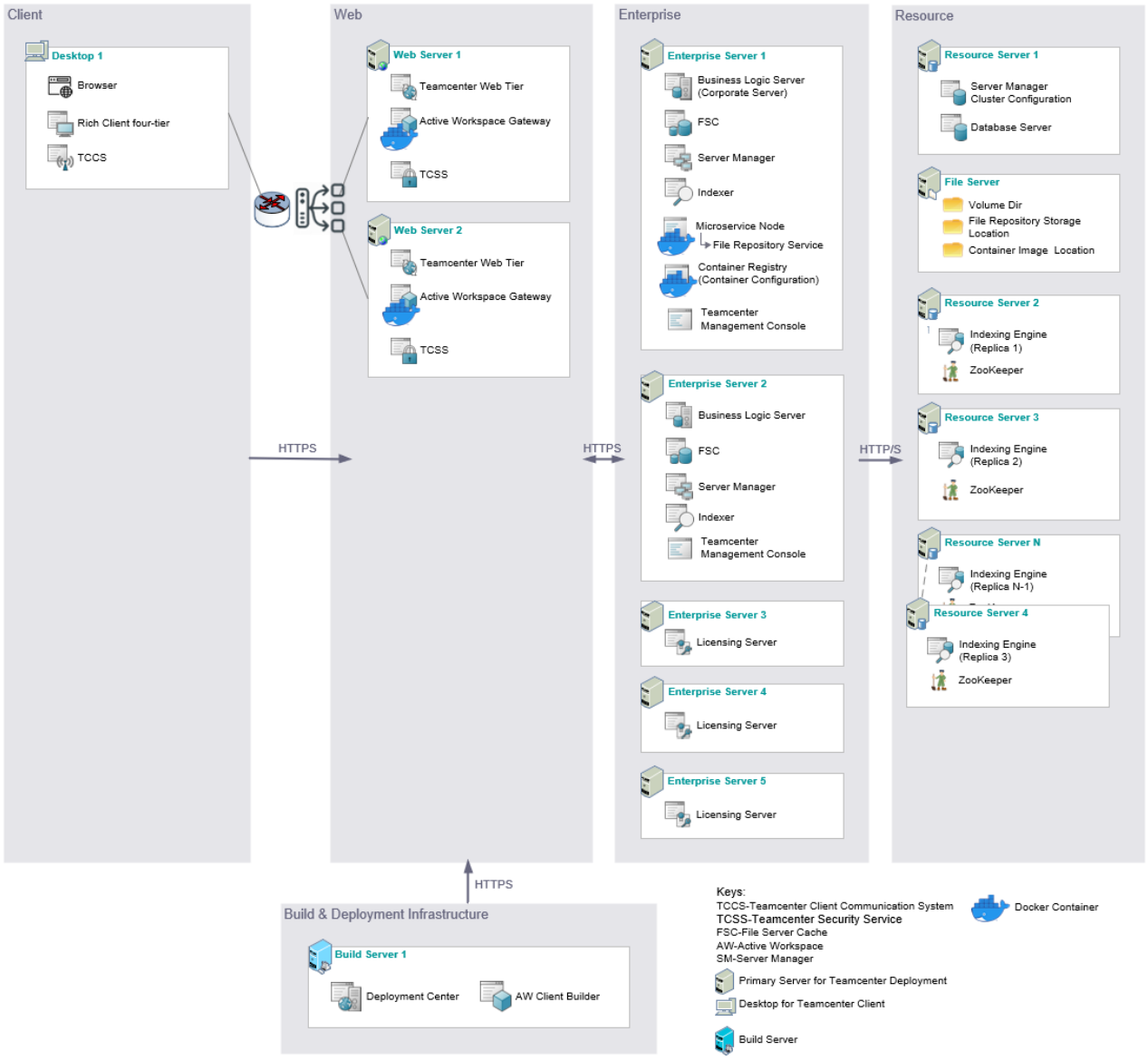
4.4 Distributed Deployment with High Availability

The following diagram shows the deployment configuration for a high availability environment with one large size resource configured as primary setup for the environment and a second, smaller resource configured as standby warm sever that always runs. The smaller resource is connected through a network switch to serve when the primary setup fails. In a virtualized environment, the secondary enterprise server could be a cold server, and the failover achieved by restoring the environment from a template.

Deployment configuration for Windows Platform



Deployment configuration for Linux Platform



A variation of this deployment would use a Licenser Server on a dedicated or common server. The License Server could support multiple environments such as development, sandbox, testing, and production.

Software/Components requirements for configuring failover

It is not mandatory to configure a failover for the entire setup. You can optimize your failover setup costs by considering your business-critical operations and software requirements. Teamcenter requires that the following components are running for you to login and perform basic functionalities:

- Web tier: Teamcenter Web tier, Active Workspace Gateway, Teamcenter Security Service (TCSS)
- Enterprise tier: Licensing Server, Business Logic Server, Server Manager, FSC, Microservice Node and Microservices (**file_repo**)
- Resource tier: Server Manger Cluster Configuration, Database Server, File Server, Indexing Engine

All other component requirements can be defined by your business-critical operations.

Achieving failover capability on Linux requires that an odd number of nodes be joined to the swarm as managers, typically three or five, so that the Docker swarm can effectively manage the swarm by majority vote. It is immaterial whether an odd or even number of nodes are joined to the swarm as workers.

Server sizing and performance guidelines

For machine sizing and scalability guidelines, refer to the Teamcenter Hardware Overview document on Support Center for more information.

Configuration Instructions

This section briefly provides instructions to configure and deploy this "Distributed Deployment with High Availability" configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and

     \quick_deploy_configurations\wntx64\Teamcenter_RA4.4_Multitier_HighAvailability_Deployment_wntx64.xml" in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.4_Multitier_HighAvailability_Deployment_lnx64.xml" in case of Linux platform.

   - and follow the instruction given in the readme "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

2. **Configure and deploy interactively using Deployment Center Client**

   Deployment Center makes configuring a High Available multi-tier deployment by offering the option to choose a distributed deployment. Use the following general instructions to configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.

   - When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
   - On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
   - It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
   - We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
   - To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.

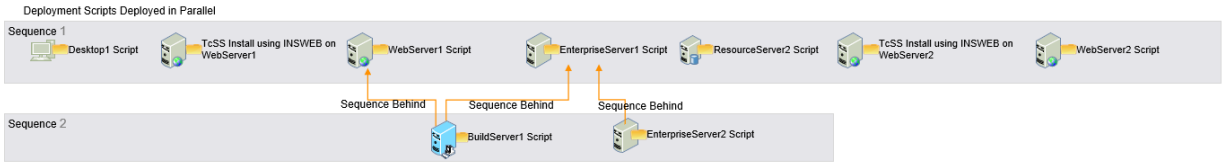
- Follow the instructions given in the Appendix Section 5.2 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.

As shown in the above configuration diagram, configure following redundancy components to serve as failover secondary components for HA (High Available) environment.

- Add “Active Workspace Gateway” and “Teamcenter Web Tier (Java EE)” from “Available Components” list and assign same web server machine to create clustered configuration. And ensure to configure 4tier connection of the “Active Workspace Gateway” component connects only to the “Teamcenter Web Tier (Java EE)” component that is configured on same web server machine.
- Teamcenter configuration supports only single endpoint connection with Teamcenter Security Service (TcSS) and for TcSS HA implementation use load balancer endpoint that can connect one of the TcSS among two TcSS.
- The “Active Workspace Client Builder” can connect to only one “Active Workspace Gateway” component to publish the client assets, so ensure to adjust the “Active Workspace Client Builder” configuration manually
- Add “Server Manager” and “Microservice Node” from “Available Components” list and assign same enterprise server machine to create clustered configuration.
- Microservice Node component deployment configuration varies based on the operating system of choice. For Linux deployment, only one Microservice Node is required to be configured. To achieve desired level of HA, increase the replica count for Service Dispatcher and Service Registry in the Microservice Node configuration to the same number of nodes in the Docker Swarm. For Windows deployments, configure one master Microservice Node. Any additional Microservice Node should be configured as worker. For Microservice Node HA, ensure to configure Service Dispatcher and Service Registry on multiple worker nodes so that they will serve as failover for the master node. Achieving failover capability on Windows requires at least two microservice nodes: a service registry, a service dispatcher, and instances of all microservices must be running on at least two nodes. By default, an instance of the service registry and service dispatcher run on the master node; additional instances can be running on any worker nodes. When installing microservice nodes via TEM, be sure to list all instances of service registry and service dispatcher.
- In this release Deployment Center supports does not support multiple components installation of “Indexer” component. For “Indexer” HA implementation use TEM Installer to install this component on to an appropriate machine as shown in the above diagram and implement with hardware switch to switch over to the standby server when primary server fails.
- As shown in the above diagram, High Availability configuration for Indexer (ResourceServer2 – RTSERVER N), at a minimum, use 3 nodes with SOLR and ZooKeeper deployed on each node. Always use odd number of nodes (3,5,7 etc.) to scale further. Here ZooKeeper acts as both an orchestrator for replicas (picks a leader automatically) and as a load balancer. For detailed instructions on Solr Cloud deploy mode, refer to “Create configsets and collections” section in the Teamcenter documentation (“Teamcenter > Documentation >Active Workspace [version] >Indexing and Search Deployment and Configuration configuration”)Edit the " Teamcenter Web Tier (Java EE) Connection(s)" on each of the Active Workspace Gateway component to keep the connection with Teamcenter Web Tier that is configured on the same machine and remove other one.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

- Desktop1 Script
- WebServer1 Script
- EnterpriseServer1 Script
- ResourceServer2 Script
- WebServer2 Script

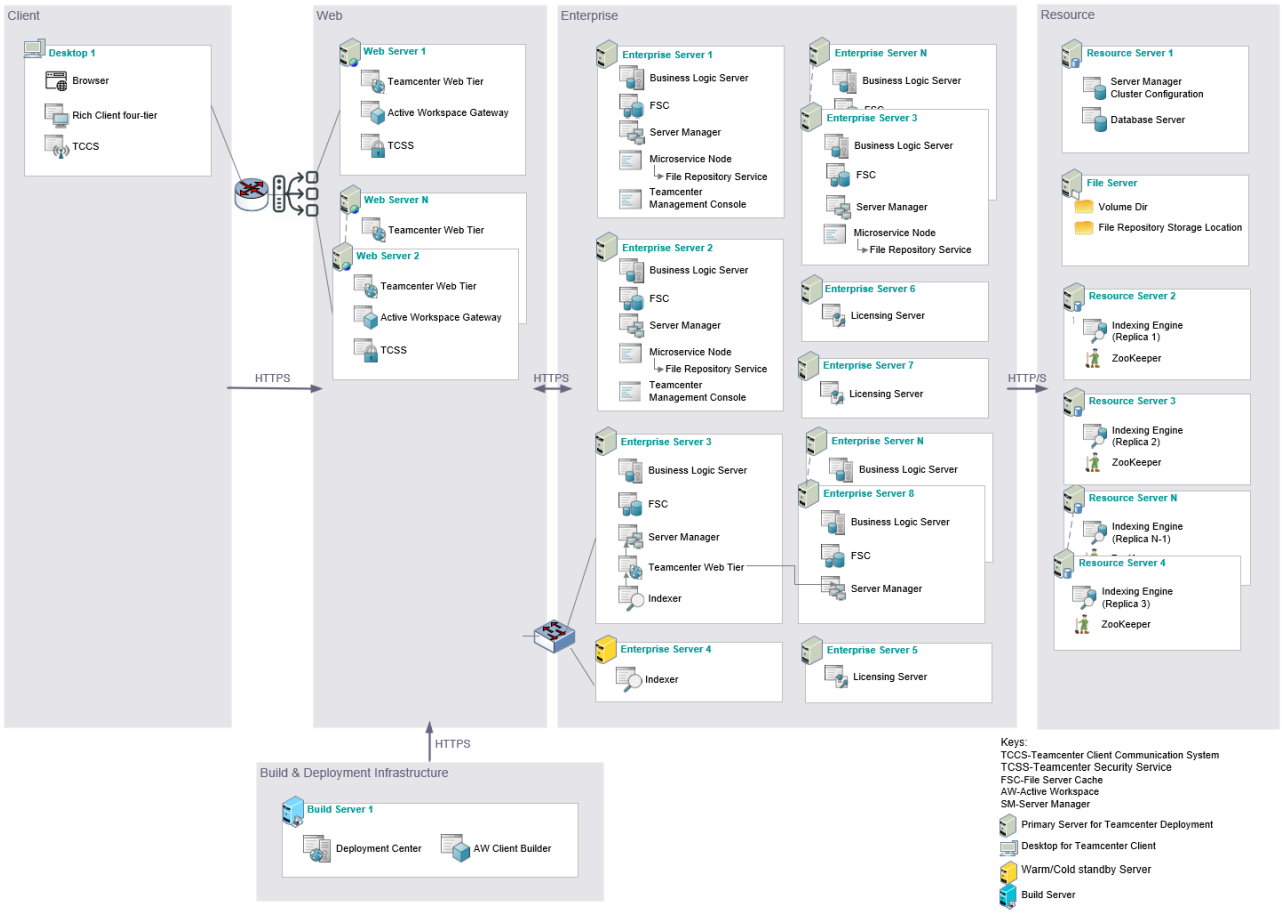
And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.

- BuildServer1 Script must be run only after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.
- EnterpriseServer2 Script must be run only after successfully running the scripts EnterpriseServer1 Script

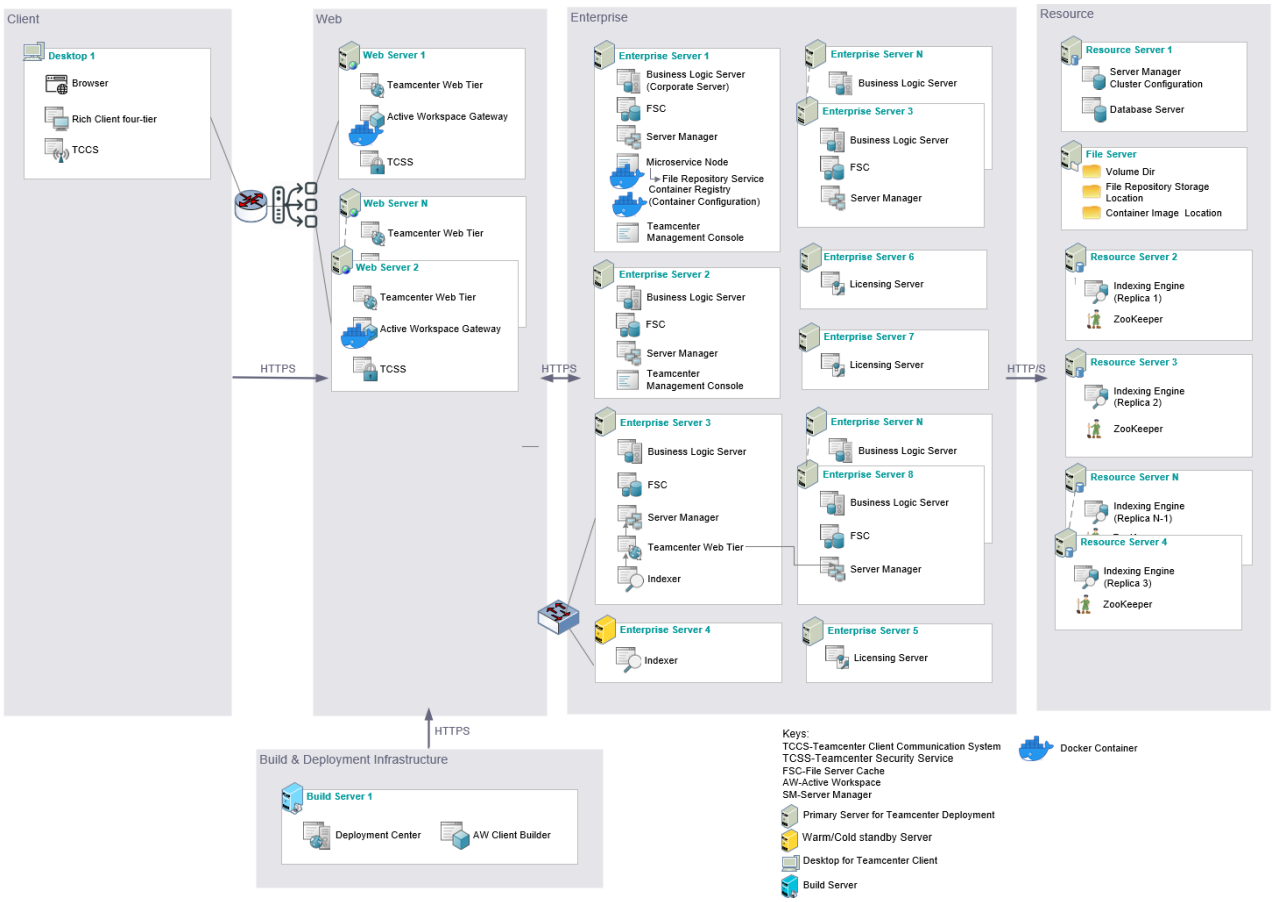
4.5 Distributed Deployment with Scalability

The following diagram shows a deployment that can be configured to scale up the environment to meet business requirements and demand. The components in each tier can be independently configured to scale up or down based on hardware specifications.

Deployment configuration for Windows Platform



Deployment configuration for Linux Platform



As shown in the diagram, the following components can be configured to deploy on multiple machines to support scaling up or down.

- Web tier: Teamcenter Web tier, Active Workspace Gateway, Teamcenter Security Service (TCSS)
- Enterprise tier: Licensing Server, Business Logic Server, Server Manager, FSC, Microservice Node and Microservices
- Resource tier: Indexing Engine

When components are configured on multiple machines, you must configure the load balancer to distribute the load across the components. There are two ways to configure the load balancer.

- **Hardware Load Balancer**: Handles high loads on all network protocols. This type of load balancing is a combination of hardware and software system. As shown in the above diagram hardware, Load Balancer is configured to act as logical service contact points for all clients of that service. Load balancers receive client requests and direct each to one active webserver.
- **Software Load Balancer**: Supports a limited set of network protocols and may lack redundancy requirements. Use the following components to configure the number of service instances that run on the machine to use the machine at its full capacity. You can also configure peer components on multiple machines to efficiently distribute the request load.

- Teamcenter Web tier
- Server Manager
- Microservices
- Indexing Engine

When File Repository Microservice configured on multiple machines, the “file repository storage location” has to be on a shared drive.

Hardware and maintenance cost can be optimized by configuring a cluster to run a group of components together on a machine. As shown in the above diagram, the following cluster can be configured to deploy components together.

- Web tier: Webserver cluster can be configured to include Teamcenter Web tier, Active Workspace Gateway, Teamcenter Security Service (TCSS)
- Enterprise tier: Business Logic cluster can be configured to include Business Logic Server, Server Manager, FSC, Microservice Node and Microservices

Dedicated hardware assignment to Indexing components for the reliable and faster performance of indexing. For example, as shown in the above diagram dedicated machine is assigned to “Indexer” and not just this but all other dependency components such as “Teamcenter Web Tier”, “Server Manager”, Business Logic Server and FSC are also configured into the same hardware so that “Teamcenter Web Tier”, “Server Manager” exclusively serves “Indexer” request without competing with functional request from user client. As noticed in the diagram the “Teamcenter Web Tier” component is configured into the Indexer machine which is in the Enterprise tier and it is ok to have it in the Enterprise tier in this case since it dedicated to Indexer and communication takes place within the machine.

Server sizing and performance guidelines

For machine sizing and scalability guidelines, refer to the Teamcenter Hardware Overview document on Support Center for more information.

Configuration Instructions

This section briefly provides instructions to configure and deploy this “Multitier Deployment with Scalability” configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.

     \quick_deploy_configurations\wntx64\Teamcenter_RA4.5_Multitier_Scalablility_Deployment_wntx64.xml” in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.5_Multitier_Scalablility_Deployment_lnx64.xml” in case of Linux platform.

   - and follow the instruction given in the readme “\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt”

2. **Configure and deploy interactively using Deployment Center Client**

Deployment Center makes configuring a scalable multi-tier deployment straightforward by offering the option to choose a distributed deployment. Use the following general instructions to configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.

- When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
- On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
- It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
- We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
- To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
- Follow the instructions given in the Appendix Section 5.2 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.

As shown in the above configuration diagram, configure following redundancy components to scale or serve as failover environment. The number of these redundancy cluster configuration is based on the business needs and scalability considerations.

- Add “Active Workspace Gateway” and “Teamcenter Web Tier (Java EE)” from “Available Components” list and assign same web server machine to create clustered configuration. And ensure to configure 4tier connection of the “Active Workspace Gateway” component connects only to the “Teamcenter Web Tier (Java EE)” component that is configured on same web server machine.
- Teamcenter configuration supports only single endpoint connection with Teamcenter Security Service (TcSS) and for TcSS load balancing/HA implementation use load balancer endpoint that can connect one of the TcSS among two TcSS.
- The “Active Workspace Client Builder” can connect to only one “Active Workspace Gateway” component to publish the client assets, so ensure to adjust the “Active Workspace Client Builder” configuration manually
- Add “Server Manager” and “Microservice Node” from “Available Components” list and assign same enterprise server machine to create clustered configuration.
- Microservice Node component deployment configuration varies based on the operating system of choice. For Linux deployment, only one Microservice node is required to be configured. To achieve desired level of HA, increase the replica count for Service Dispatcher and Service Registry in the Microservice Node configuration to the same number of nodes in the Docker Swarm. For Windows deployment, configure one master Microservice Node. Any additional Microservice Node should be configured as worker. For Microservice Node HA, ensure to configure Service Dispatcher and Service Registry on multiple worker nodes so that they will serve as failover for the master node.

In this case File Repository Microservices are configured on multiple machines, the “file repository storage location” must be configured on a shared drive.

- In this release Deployment Center supports does not support multiple components installation of “Indexer” component. For “Indexer” HA implementation use TEM Installer to install this component on to an appropriate machine as shown in the above diagram and implement with hardware switch to switch over to the standby server when primary server fails.
- As shown in the above diagram, High Availability configuration for Indexer (ResourceServer2 – RTSERVER N), at a minimum, use 3 nodes with SOLR and ZooKeeper deployed on each node. Always use odd number of nodes (3,5,7 etc.) to scale further. Here ZooKeeper acts as both an orchestrator for replicas (picks a leader automatically) and as a load balancer. For detailed instructions on Solr Cloud deploy mode, refer to “Create configsets and collections” section in the Teamcenter documentation (“Teamcenter > Documentation >Active Workspace [version] >Indexing and Search Deployment and Configuration configuration”)
- The one “Teamcenter Management Console” component is sufficient to monitor all Server Manager/Teamcenter Webtier components but to achieve HA, configure another “Teamcenter Management Console” on the 2nd enterprise server. Ensure that there are no more than 2 of them.
- Edit the "Teamcenter Web Tier (Java EE) Connection(s)" on each of the Active Workspace Gateway component to keep the connection with Teamcenter Web Tier that is configured on the same machine and remove other one.
- Edit the "Teamcenter Web Tier (Java EE) Connection(s)" on the Visualization Data Server to remove the connection with Teamcenter Web Tier that is configured on the Indexer machine.

As shown in the above configuration diagram, assign dedicated machine to “Indexer” component for the reliable and faster performance and not just this but configure the same machine to all other dependency components such as “Teamcenter Web Tier”, “Server Manager”, Business Logic Server and FSC so that “Indexer” connects and communicate within the machine using HTTP and “Teamcenter Web Tier”, “Server Manager” are configured to serves “Indexer” request exclusively.

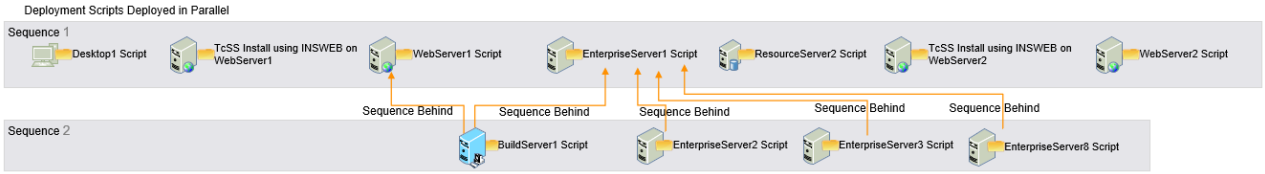
Configure the Indexer to have dedicated Teamcenter Web Tier and Server Manager for faster and reliable Indexing performance.

- Edit the "Teamcenter Web Tier (Java EE) Connection(s)" on the Indexer component to keep the connection with Teamcenter Web Tier that is configured on the same machine and remove other one.
- Edit the "Server Manager Connection(s)" on the Teamcenter Web Tier component that is configured to the Indexer machine to keep the connection with Server Manager that is configured on the same machine and remove the other one.
- Edit the "Server Manager Connection(s)" on each of the Teamcenter Web Tier component (other than the one that is configured on Indexer machine) to remove the connection with Server Manager that is configured on the Indexer machine.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows

the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

- Desktop1 Script
- WebServer1 Script
- EnterpriseServer1 Script
- ResourceServer2 Script
- WebServer2 Script

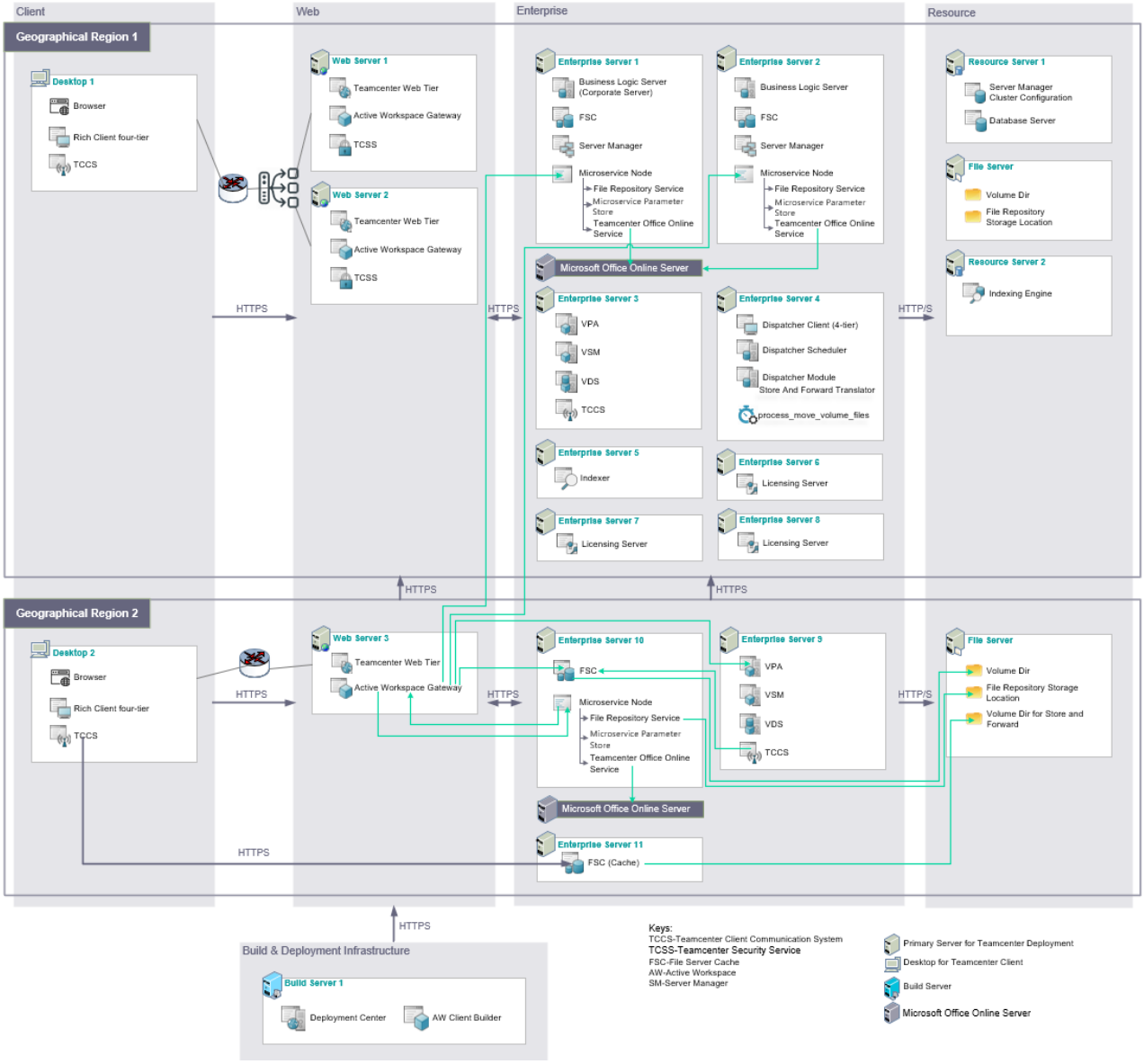
And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.

- BuildServer1 Script must be run only after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.
- EnterpriseServer2, EnterpriseServer3 and EnterpriseServer8 Scripts must be run only after successfully running the scripts EnterpriseServer1 Script

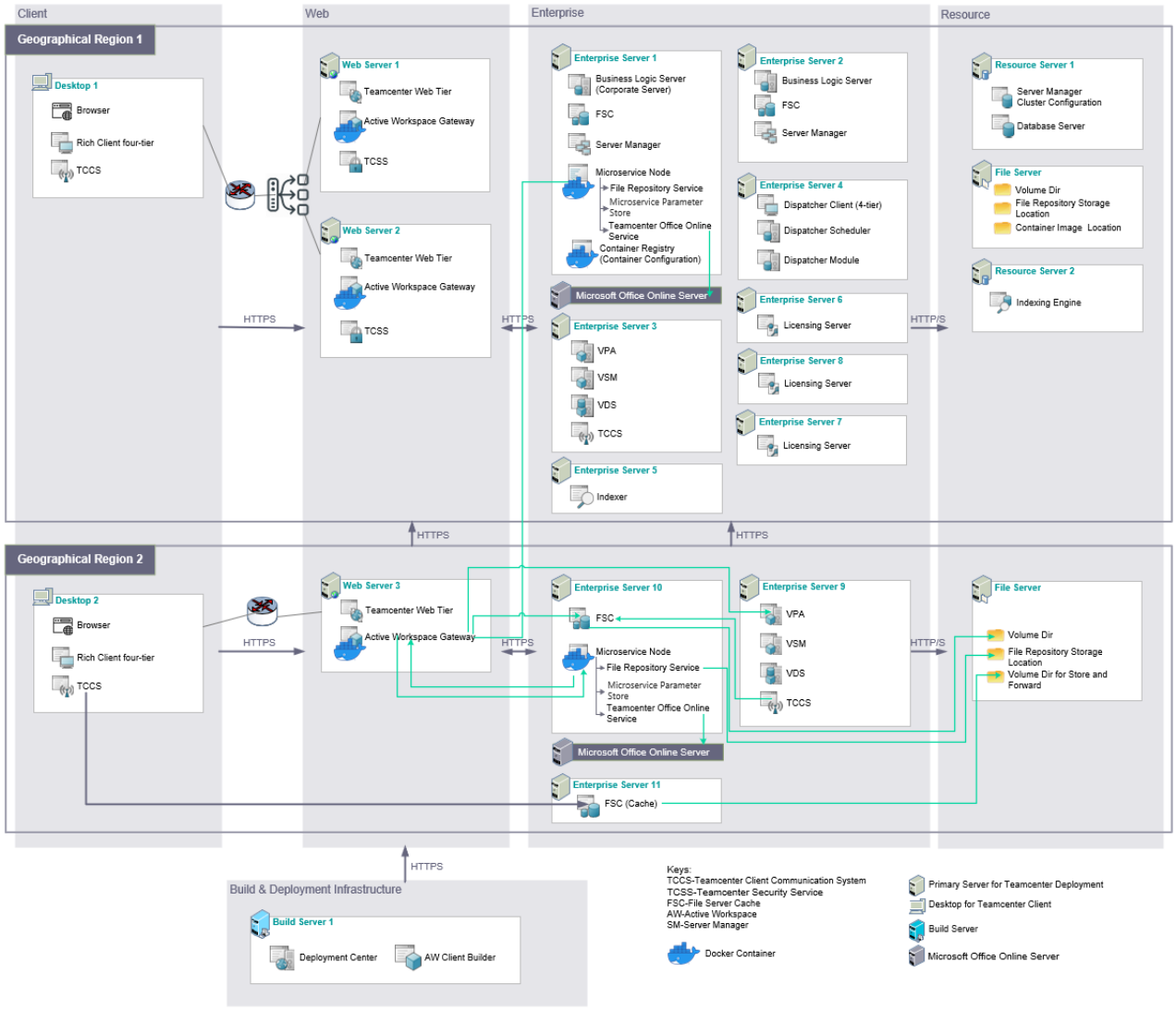
4.6 Distributed Global Deployment

The following diagram illustrates a global deployment strategy that can lower costs by configuring a centralized data infrastructure in a single region and distributing Client & Web tiers with file management system across other geographical regions.

Deployment configuration for Windows Platform



Deployment configuration for Linux Platform



Forward deployment

This section provides detailed instructions for forward deploying Teamcenter office online microservice and the corresponding required components in a different region for the security regulations and performance reasons.

The scenario considered in this deployment is that Teamcenter enterprise and its relevant components already exist in the geographical region 1, and then to forward deploy the following components in geographical region 2 along with local volume and File Repository Storage directories.

- Web tier: Teamcenter Web Tier, Active workspace gateway
- Enterprise tier:
  - File Server Cache (FSC)
  - Microservice Node
    - Teamcenter Office Online Service
    - File repository service

- Microsoft Office Online Server
- Visualization Server Manager, Visualization Pool Assigner and Visualization Data Server

Deployment of Store and Forward Volume

Default local volumes are temporary local volumes that allow files to be stored locally before they are automatically transferred to the destination volume. This functionality improves end-user file upload times from clients by uploading files to a temporary volume and it also referred to as store and forward functionality.

The above diagram also illustrates a deployment of Store and Forward volume, the overall configuration of Store and Forward volume involves following things.

- Configuring and deploying FSC/Dispatcher/Store and Forward translator for Store and Forward Volume
- Configuring and scheduling a cron job or scheduled task to run process_move_volume_files bat/sh script to move volumes
- Enable Store & Forward Preferences using Active Workspace/Rich Client Admin application
- Configure a local volume at the remote site using Active Workspace/Rich Client Admin application

For more detailed instruction, refer to the following sections in Teamcenter Help Guide.

Teamcenter system administration → File Management System →Administering FMS → Administering Volumes → Default Local Volumes

Server sizing and performance guidelines

For machine sizing and scalability guidelines, refer to the Teamcenter Hardware Overview document on Support Center for more information.

Configuration Instructions

This section briefly provides instructions to configure and deploy this “Distributed Global Deployment” configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.

     \quick_deploy_configurations\wntx64\Teamcenter_RA4.6_Multitier_Global_Deployment_wntx64.xml” in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.6_Multitier_Global_Deployment_lnx64.xml” in case of Linux platform.

- and follow the instruction given in the readme "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

2. Configure and deploy interactively using Deployment Center Client

Deployment Center makes configuring a multi-tier global deployment by offering the option to choose a distributed deployment. Use the following general instructions to configure a multi-tier distributed deployment in the 1st geographical region. Refer to Teamcenter deployment guide for detailed instructions.

- When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
- On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
- It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
- We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
- To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
- Follow the instructions given in the Appendix Section 5.2 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.

As shown in the above configuration diagram, configure centralized environment with the following redundancy components to scale or serve as failover. The number of these redundancy cluster configuration is based on the business needs and scalability considerations.

- Add "Active Workspace Gateway" and "Teamcenter Web Tier (Java EE)" from "Available Components" list and assign same web server machine to create clustered configuration. And ensure to configure 4tier connection of the "Active Workspace Gateway" component connects only to the "Teamcenter Web Tier (Java EE)" component that is configured on same web server machine.
- Teamcenter configuration supports only single endpoint connection with Teamcenter Security Service (TcSS) and for TcSS load balancing/HA implementation use load balancer endpoint that can connect one of the TcSS among two TcSS.
- The "Active Workspace Client Builder" can connect to only one "Active Workspace Gateway" component to publish the client assets, so ensure to adjust the "Active Workspace Client Builder" configuration manually
- Add "Server Manager" and "Microservice Node" from "Available Components" list and assign same enterprise server machine to create clustered configuration.
- Microservice Node component deployment configuration varies based on the operating system of choice. For Linux global deployment, only one Microservice Node is required to be configured per region. Each region should have its own Docker Swarm configured. A single container registry can be used to deploy all microservice images for both regions. If there is a firewall between both regions, container registry port needs to be exposed in order for the region 2 Docker Swarm to be able to access the microservice images. For Windows global deployment, one master Microservice

Node. Any additional Microservice Node should be configured as worker and make sure to select Service Dispatcher and Service Registry and specify the instance count > 0 for required microservices.

- In this release Deployment Center supports does not support multiple components installation of “Indexer” and “Indexing Engine” component. For “Indexer” and “Indexing Engine” scalability/HA implementation use TEM Installer to install these components on to an appropriate machine as shown in the above diagram.

As shown in the above configuration diagram, configure following redundancy clustered configuration in the 2nd geographical region as a forward deployment for the security regulations and performance reasons.

- Add “Active Workspace Gateway” and “Teamcenter Web Tier (Java EE)” from “Available Components” list and assign same web server machine in the 2nd geographical region to create clustered configuration. And ensure to configure 4tier connection of the “Active Workspace Gateway” component connects only to the “Teamcenter Web Tier (Java EE)” component that is configured on same web server machine.
- Teamcenter configuration supports only single endpoint connection with Teamcenter Security Service (TcSS) and for TcSS load balancing implementation use load balancer endpoint that can connect one of the TcSS of 1st geographical region.
- Add “Visualization Server Manager”, “Visualization Pool Assigner” and “Visualization Data Server” from “Available Components” list and assign same server machine in the 2nd geographical region to create clustered configuration.
- And add “Microservice Node” and “FSC” components from “Available Components” list and assign same server machine for both in the 2nd geographical region.
- Microservice Node component deployment configuration varies based on the operating system of choice. For Linux global deployment, a Microservice Node is required to be configured per region. For the region 2, configure a Docker Swarm separate from the region 1. A single container registry can be used to deploy all microservice images for both regions. If there is a firewall between both regions, the container registry port needs to be exposed in order for the region 2 Docker Swarm to be able to access the microservice images.

  For Windows deployments, configure a dedicated worker node in the region 2 which is isolated from the region 1. On the Microservice Node dedicated to region 2, select Service Dispatcher, Service registry and specify the instance count > 0 for File Repository, Microservice Parameter Store, Office Online microservice microservices. Additionally, specify Microservice Office online server which is deployed in region 2.

By default, Deployment Center automatically connects Active Workspace Gateway/Teamcenter Web Tier with all instances of Microservice Node/FSC/Visualization Components that are available in the environment. In order to accomplish the forward deployment, it is required to adjust the connections as shown in the above diagram. Follow the instructions given below to edit connections on the respective component in the 2nd geographical region to accomplish the forward deployment.

- Select 2nd geographical region “Active Workspace Gateway” component from the component table, click on “Show all parameters” eye button on the component panel and edit the following connections,

    - edit “Visualization Pool Assigner Connection(s)” to connect the “Visualization Pool Assigner” of the 2nd geographical region.
    - By default, Deployment Center selects “Use as Bootstrap URLs” option, keep this option for now and edit “FSC Connection(s) “to connect to “FSC” of the 2nd geographical region and then switch option to “Use Assigned FSC URLs” so that 2nd geographical region “Active Workspace Gateway” will only connect to the “FSC” that is just configured.
- Select 2nd geographical region “Microservice Node” component from the component table, click on “Show all parameters” eye button on the component panel and edit “Active Workspace Gateway Connection(s)” to connect “Active Workspace Gateway” of the 2nd geographical region.
- Select 2nd geographical region “TCCS” component on the Visualization Server from the component table, click on “Show all parameters” eye button on the component panel and edit “FSC Connection(s)” to connect “FSC” of the 2nd geographical region.
- Configure 2nd geographical region “FSC” with local volume in the 2nd geographical region.

Post deployment configuration

On successful deployment, make the required following forward deployed gateway routing change for office online microservice requests properly routed to the forward deployed Teamcenter Office Online microservice.

Manually edit 2nd geographical region Service Dispatcher’s json/yml file and define an environment variable “SD_FORWARD_URLS” and whose value is a comma separated list of Active Workspace Gateway URLs in the 1st geographical region.

For example, SD_FORWARD_URLS=http://gateway1:3000,http://gateway2:3000

How does this work? If the forward deploy’s Service Dispatcher does NOT find the microservice locally running, it will forward the request on to a randomly selected one from the list defined in “SD_FORWARD_URLS”.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.

The following deployment scripts listed as Sequence 1 can be run parallel in any order,

- Desktop1 Script
- WebServer1 Script
- ResourceServer2 Script

- EnterpriseServer1 Script
- EnterpriseServer3 Script
- EnterpriseServer4 Script
- EnterpriseServer5 Script
- EnterpriseServer9 Script
- EnterpriseServer10 Script
- WebServer2 Script
- Desktop2 Script, WebServer3 Script

And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.+

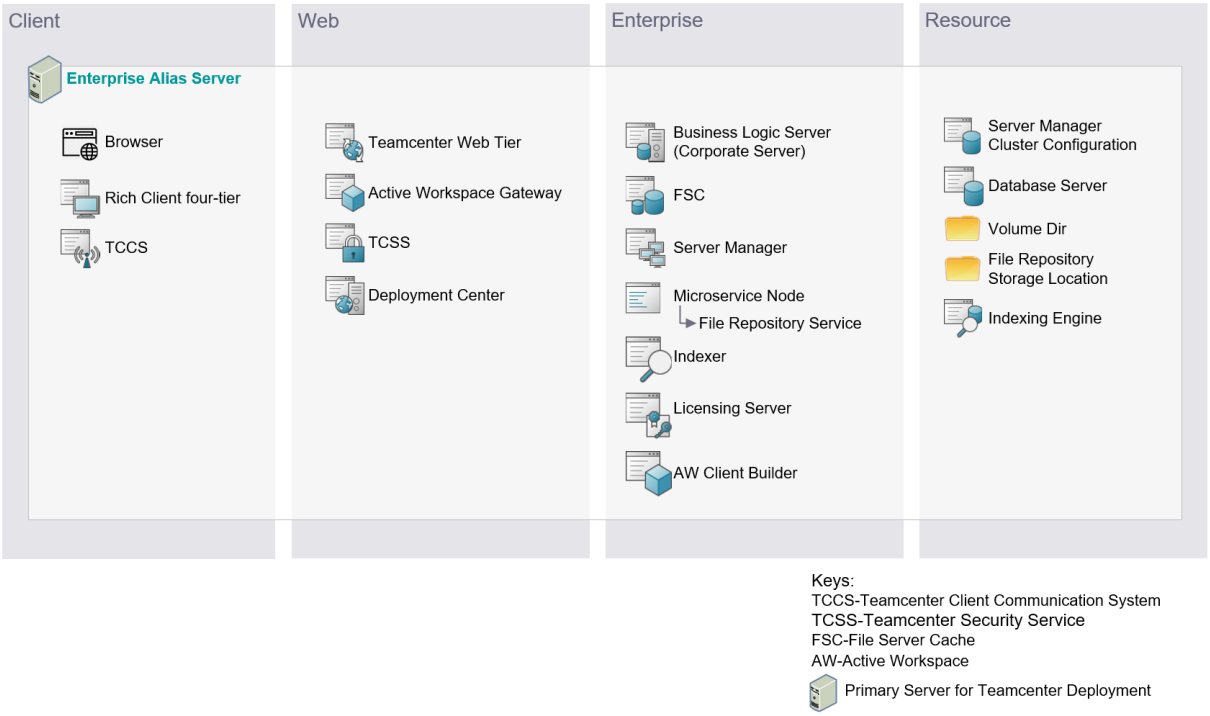
- BuildServer1 Script must be run only after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.
- EnterpriseServer2 Script must be run only after successfully running the scripts EnterpriseServer1 Script

4.7 One Box Deployment for Environment Cloning

Cloning an environment from another environment is very commonly and frequently performed operation in the industry, and it is normally used for setting up Development/UAT environment for performing development or testing upgrade/patching.

In order to clone a correctly functioning environment from an existing environment, it is required to configure and deploy the original environment in a particular way. The environment cloning is achieved using the method of configuring the deployment in a specific way in the Deployment Center in combination with virtualization. So, it is recommended to use virtual machine and one box deployment to prepare an environment for cloning.

In this section we have considered “Minimal One Box Deployment” configuration example that is documented in the section 4.1, however you may even choose “All in One Box Deployment” configuration example that is documented in the section 4.2. The following diagram shows all the required deployment architecture components for this one server deployment. They are all configured to deploy on single server machine for cloning purpose.



Configuration Instructions

This section briefly provides instructions to configure and deploy this “One Box Deployment” configuration on virtual machine for Cloning purpose.

1. **Prepare the hardware for Teamcenter/Active Workspace deployment**
   - Create Virtual Machine using any virtualization solution available in the business
   - Modify the Hostname for a Virtual Machine as per the IT requirement.
2. **Configure and deploy Teamcenter/Active Workspace On virtual machine using Alias Host Name**
   2.1. Via Quick Deploy command line utility

- Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.
  \quick_deploy_configurations\wntx64\Teamcenter_RA4.7_Onebox_Deployment_For_Cloning_wntx64.xml" in case of Windows platform and
  \quick_deploy_configurations\lnx64\Teamcenter_RA4.7_Onebox_Deployment_For_Cloning_lnx64.xml" in case of Linux platform.
- and follow the instruction given in the readme
  "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

2.2. Via interactively using Deployment Center Client

A single server deployment is Deployment Center's default configuration. On selection of Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration and automatically selects the basic Teamcenter/Active Workspace application and all the required components.

When you assign a machine to one of the components, ensure to specify the Alias Host Name instead the physical host name of the virtual machine. The Deployment Center automatically configures the same machine to all the components except the Database Server and Licensing Server for which you have the option to assign the same Alias Host Name or a different machine.

The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

3. On successful deployment of Teamcenter/Active Workspace, clone a Virtual Machine to a Template as per the instructions provided by the virtualization solution.
4. Clone a Teamcenter environment from the Virtual Machine Template. Follow the instructions provided by the virtualization solution to clone a Virtual Machine from the Template that was captured in the step 3.
5. Add an Alias Host Name in the /etc/hosts file
   - From a virtual machine that has access to your storage system, edit the host files (depending on OS: Windows under computer properties, Linux you would need to edit the /etc/hostname & /etc/hosts files)
   - Add the following line to the /etc/hosts file:***IP_address host_name aliases***
     IP_address is the IP address of the host., *host_name* is the name of the host., *aliases* are the alias names for the host.

   Example

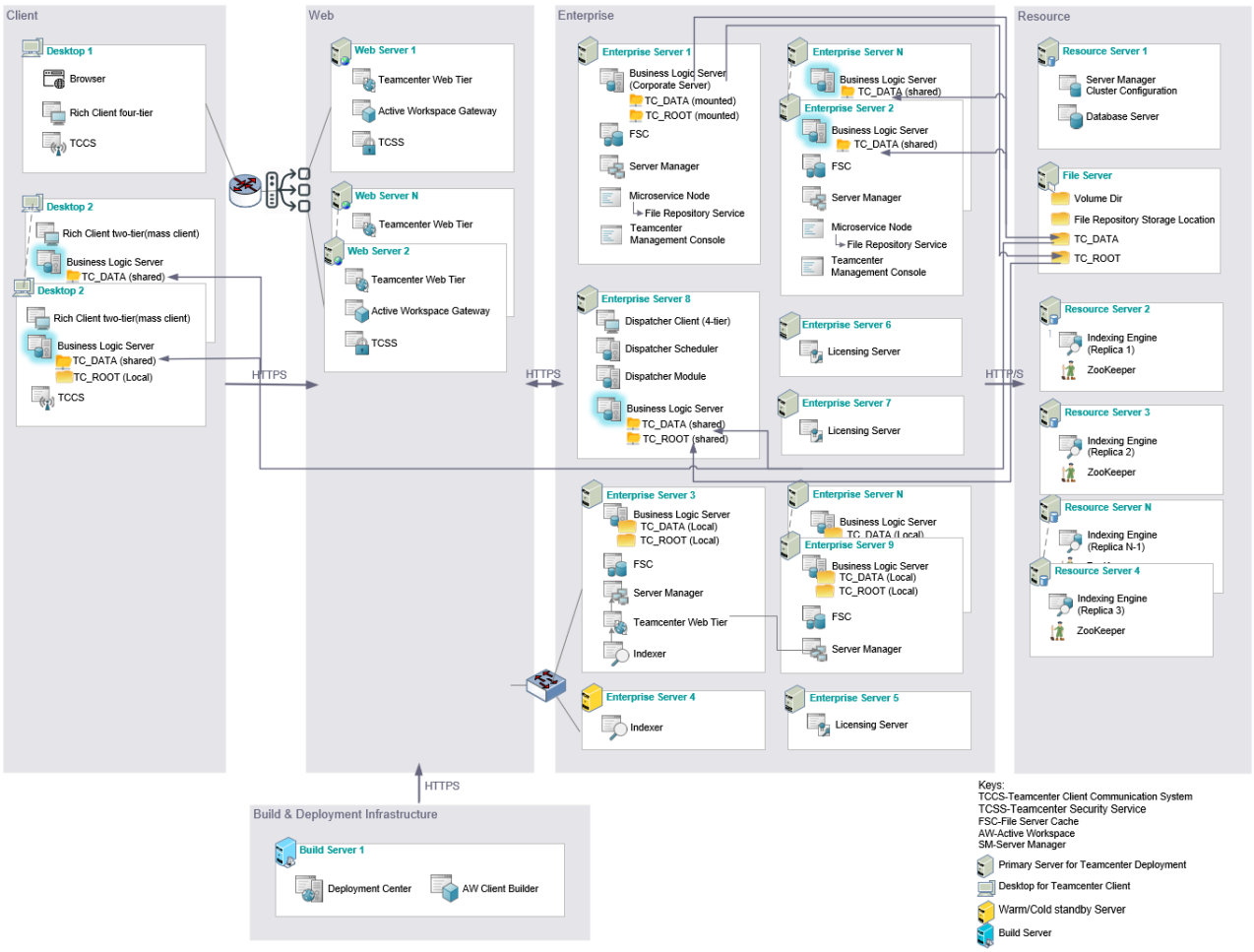
   To add a host name, myhost, with an IP address 192.0.2.16, add the following line in the /etc/hosts file:

   192.0.2.16 myhost myAliasHost

   Here myAliasHost is the alias name for myhost. Restart all Teamcenter service and launch the Active Workspace Client/Rich Client to experience the Teamcenter functionality.

4.8 Distributed deployment using shared TC_ROOT and TC_DATA

The following diagram illustrates scalability deployment that can be configured to use shared TC_ROOT/TC_DATA instead of deploying Teamcenter Server with its server binaries and data on to the local machine.



Following are the known constraints to be aware of with shared/mounted directory.

- The operating system credentials are necessary to connect mounted drives/shared directory.
- If any services configured to start automatically, they may fail to start if mounted/shared directory are not available at the startup.
- The Server/Clients that use shared directory need to have read/write permissions.

Keep the following best practices in mind.

- Avoid sharing the TC_ROOT with Rich Client two tier Teamcenter Server as it is going to put heavy payload on network in pulling gigabytes of binaries across the network.
- Evaluate sharing of TC_ROOT with additional Teamcenter Server. If each of the Teamcenter Server configured to spin hundreds of server instances, then multiples of these could lead to hundreds of

server instances that will try to connect to shared directory . This may cause the performance degradation depending on the hardware/file system/network latency.

- Share TC_DATA directory mostly with all Teamcenter Server configured with Server Manager, Rich Client two tier and Dispatcher Module as data/binaries pulling across tier/network will be smaller in size.

Configuration Instructions

This section briefly provides instructions to configure and deploy "Multitier Deployment using Shared TC_ROOT and TC_DATA" configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped as part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.

     \quick_deploy_configurations\wntx64\Teamcenter_RA4.8_Shared_RootDataDir_Deployment_wntx64.xml" in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.8_Shared_RootDataDir_Deployment_lnx64.xml" in case of Linux platform.

   - Follow the instruction given in the readme. "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

2. **Configure and deploy interactively using Deployment Center Client**

   Deployment Center makes configuring multi-tier deployment with shared TC_ROOT/TC_DATA dir straightforward by offering the option to choose a distributed deployment. Use the following general instructions to configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.

   - When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
   - On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
   - It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
   - We highly recommend that you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
   - To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
   - Follow the instructions given in the Appendix Section 5.2 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.

As shown in the above configuration diagram, configure the following redundancy components to scale or serve as failover environment. The number of these redundancy cluster configuration is based on the business needs and scalability considerations.

- Add “Active Workspace Gateway” and “Teamcenter Web Tier (Java EE)” from “Available Components” list and assign the same web server machine to create clustered configuration. Ensure to configure 4 tier connection of the “Active Workspace Gateway” component connects only to the “Teamcenter Web Tier (Java EE)” component that is configured on same web server machine.
- Teamcenter configuration supports only single endpoint connection with Teamcenter Security Service (TcSS) and for TcSS load balancing/HA implementation use load balancer endpoint that can connect one of the TcSS among two TcSS.
- The “Active Workspace Client Builder” can connect to only one “Active Workspace Gateway” component to publish the client assets, so ensure to adjust the “Active Workspace Client Builder” configuration manually
- Add “Server Manager” and “Microservice Node” from “Available Components” list and assign same enterprise server machine to create clustered configuration.
- Microservice Node component deployment configuration varies based on the operating system of choice. For Linux deployment, only one Microservice node is required to be configured. To achieve desired level of HA, increase the replica count for Service Dispatcher and Service Registry in the Microservice Node configuration to the same number of nodes in the Docker Swarm. For Windows deployment, configure one master Microservice Node. Any additional Microservice Node should be configured as worker. For Microservice Node HA, ensure to configure Service Dispatcher and Service Registry on multiple worker nodes so that they will serve as failover for the master node.
- In this release Deployment Center does not support multiple components installation of “Indexer” component. For “Indexer” HA implementation use TEM Installer to install this component on to an appropriate machine as shown in the above diagram and implement with hardware switch to switch over to the standby server when primary server fails.
- As shown in the above diagram, High Availability configuration for Indexer (ResourceServer2 – RTSERVER N), at a minimum, uses 3 nodes with SOLR and ZooKeeper deployed on each node. Always use odd number of nodes (3,5,7 etc.) to scale further. Here ZooKeeper acts as both an orchestrator for replicas (picks a leader automatically) and as a load balancer. For detailed instructions on Solr Cloud deploy mode, refer to “Create configsets and collections” section in the Teamcenter documentation (“Teamcenter > Documentation >Active Workspace [version] >Indexing and Search Deployment and Configuration configuration”)
- Edit the "Teamcenter Web Tier (Java EE) Connection(s)" on each of the Active Workspace Gateway component to keep the connection with Teamcenter Web Tier that is configured on the same machine and remove other one.
- Edit the "Teamcenter Web Tier (Java EE) Connection(s)" on the Visualization Data Server to remove the connection with Teamcenter Web Tier that is configured on the Indexer machine.

As shown in the above configuration diagram, assign dedicated machine to “Indexer” component for the reliable and faster performance and not just this but configure the same machine to all other dependency components such as “Teamcenter Web Tier”, “Server Manager”, Business Logic Server

and FSC so that “Indexer” connects and communicate within the machine using HTTP and “Teamcenter Web Tier”, “Server Manager” are configured to serves “Indexer” request exclusively.

Configure the Indexer to have dedicated Teamcenter Web Tier and Server Manager for faster and reliable Indexing performance.

- Edit the "Teamcenter Web Tier (Java EE) Connection(s)" on the Indexer component to keep the connection with Teamcenter Web Tier that is configured on the same machine and remove other one.
- Edit the "Server Manager Connection(s)" on the Teamcenter Web Tier component that is configured to the Indexer machine to keep the connection with Server Manager that is configured on the same machine and remove the other one.
- Edit the "Server Manager Connection(s)" on each of the Teamcenter Web Tier component (other than the one that is configured on Indexer machine) to remove the connection with Server Manager that is configured on the Indexer machine.

Configure Teamcenter Business Logic Server to use shared TC_ROOT directory

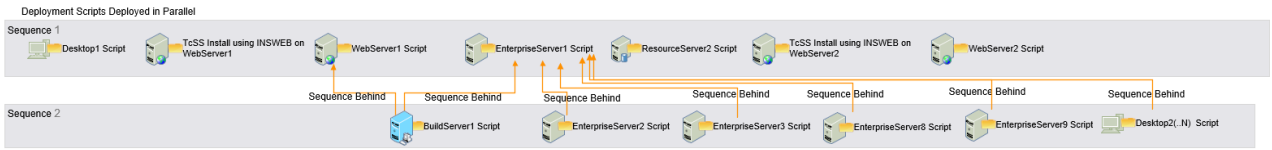
- Select “Business Logic Server” component from Selected Components table.
- To use the shared TC_ROOT directory, select “Shared TC_ROOT?” checkbox under the “Shared Teamcenter Settings”
- Choose the “Use Shared Teamcenter Root Path” option if to specify the exact shared Teamcenter root path directory to use or choose “Use Environment Variable TC_ROOT_SHARED during deployment” option if to use the shared dir value specified from the TC_ROOT_SHARED env variable.

Configure Teamcenter Business Logic Server to use shared TC_DATA directory.

- Select “Business Logic Server” component from Selected Components table.
- To use the shared TC_DATA directory, select “Shared TC_DATA?” checkbox under the “Shared Teamcenter Settings”
- Choose the “Use Shared Teamcenter Data Path” option if to specify the exact shared Teamcenter root path directory to use or choose “Use Environment Variable TC_DATA_SHARED during deployment” option if to use the shared dir value specified from the TC_DATA_SHARED env variable.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

- Desktop1 Script

- WebServer1 Script
- EnterpriseServer1 Script
- ResourceServer2 Script
- WebServer2 Script

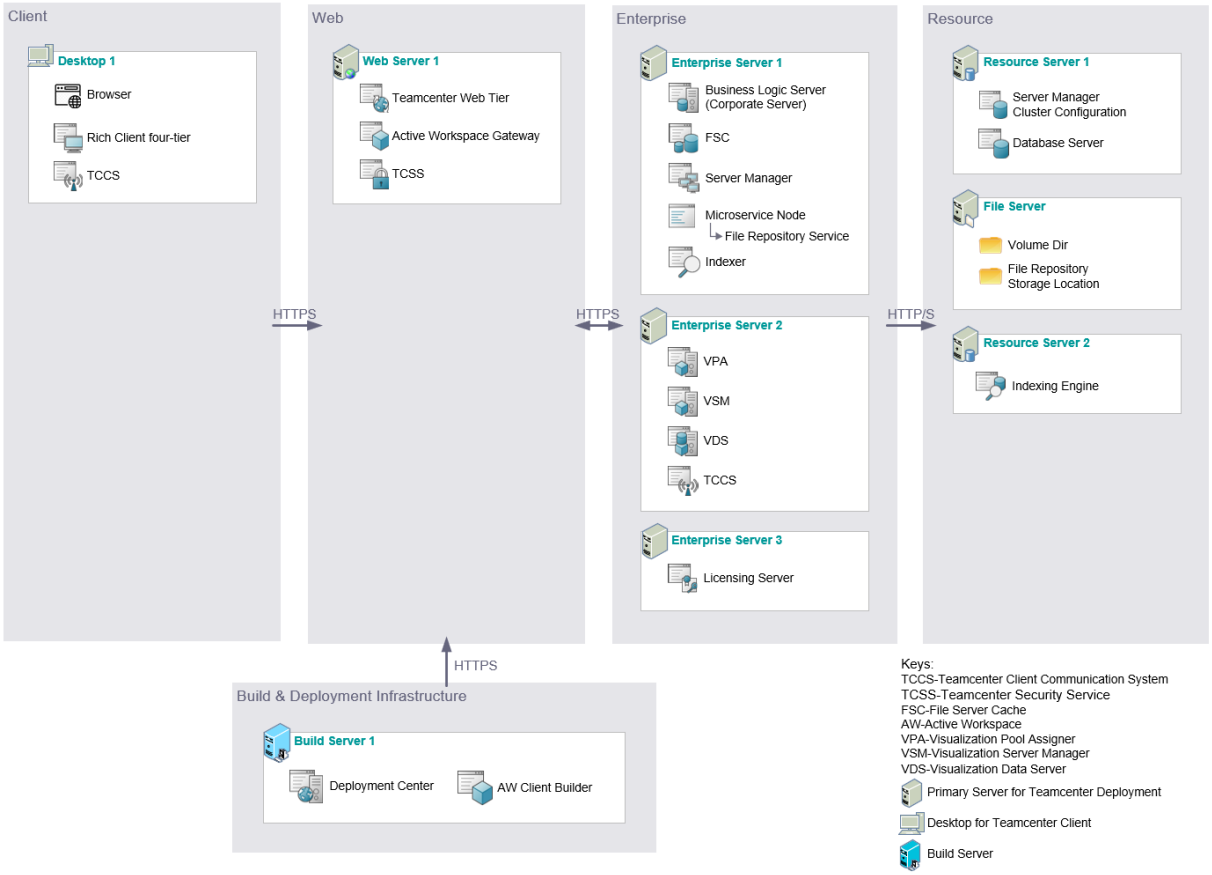
the following deployment scripts listed as Sequence 2 must be run after Sequence 1 deploy scripts execution are completed. The scripts mentioned in Sequence 2 can be run among them in parallel in any order.

- BuildServer1 Script must be run only after successfully running WebServer1 Script and EnterpriseServer1 Script.
- EnterpriseServer2, EnterpriseServer3 and EnterpriseServer8, EnterpriseServer9, Desktop2(…N) Scripts must be run only after successfully running EnterpriseServer1 Script

4.9 Optional Software Components

4.9.1 Multi-tier with 3D Visualization

The following diagram shows deployment configuration specific to 3D Visualization Deployment on a multi-tier environment.



Configuration Instructions

This section briefly provides instructions to configure and deploy this “Multi-tier with 3D Visualization” configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.

     \quick_deploy_configurations\wntx64\Teamcenter_RA4.9.1_Multitier_3DVisualization_Deployment_wntx64.xml” in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.9.1_Multitier_3DVisualization_Deployment_lnx64.xml” in case of Linux platform.

- and follow the instruction given in the readme "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

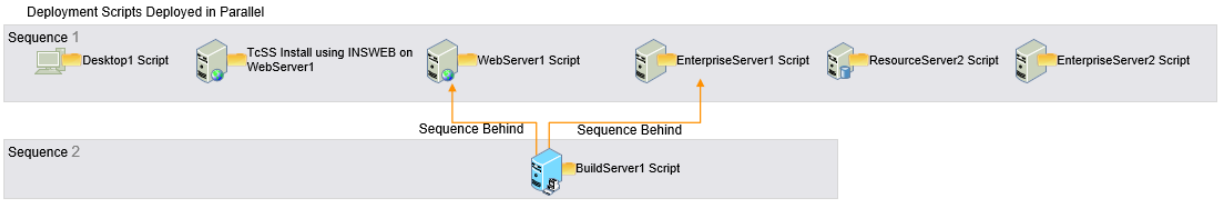
2. Configure and deploy interactively using Deployment Center Client

Deployment Center makes configuring a multi-tier deployment with 3D Visualization Server is straightforward by offering the option to choose a distributed deployment. Use the following general instructions to configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.

- When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
- On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
- It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
- We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
- To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
- Select application "3D Visualization" on the **"3 Applications"** tab to install Visualization for 3D Rendering support.
- Add "Visualization Data Server" from "Available Components" list and assign same server machine to "Visualization Server Manager", "Visualization Pool Assigner" and "Visualization Data Server" to create clustered configuration.
- Follow the instructions given in the Appendix Section 5.2 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.
- The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

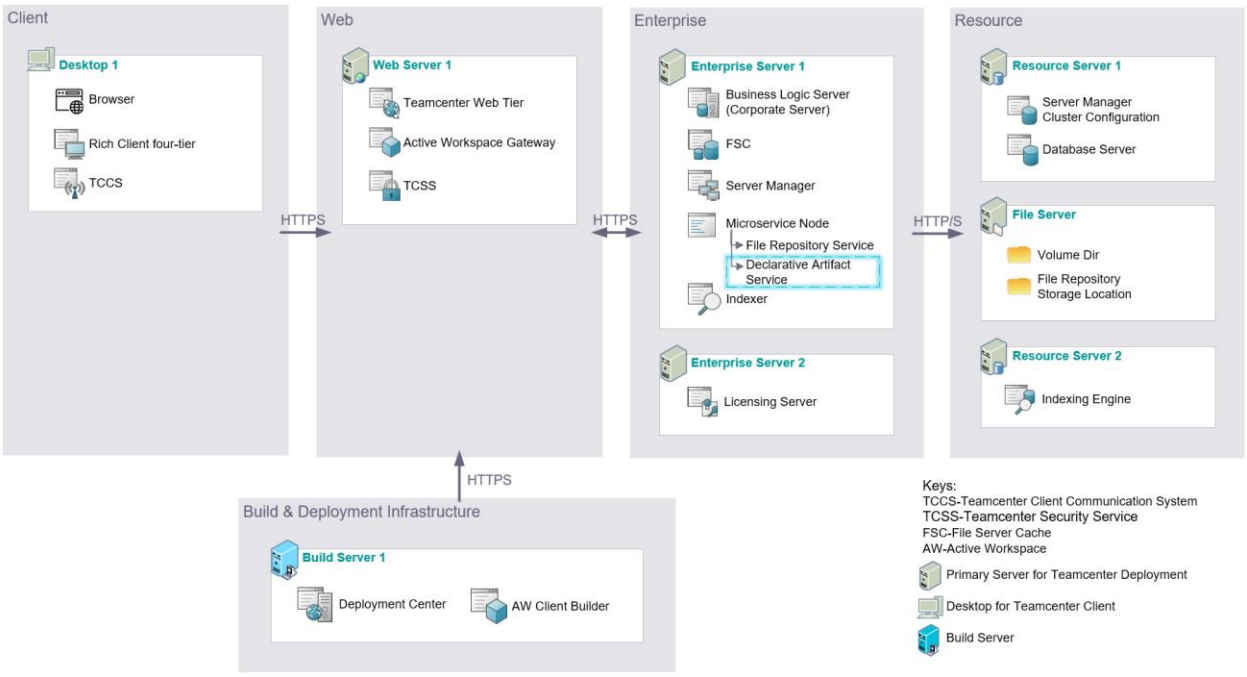
- Desktop1 Script
- WebServer1 Script
- ResourceServer2 Script
- EnterpriseServer1 Script
- EnterpriseServer2 Script

And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.

- BuildServer1 Script must be run after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.

4.9.2 Multi-tier with Active Workspace UI Builder

The following diagram shows deployment configuration specific to Active Workspace UI Builder Deployment on a multi-tier environment.



Configuration Instructions

This section briefly provides instructions to configure and deploy this "Multi-tier with Active Workspace UI Builder" configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.

     \quick_deploy_configurations\wntx64\Teamcenter_RA4.9.2_Multitier_AWUIBuilder_Deployment_wntx64.xml" in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.9.2_Multitier_AWUIBuilder_Deployment_lnx64.xml" in case of Linux platform.

   - and follow the instruction given in the readme "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

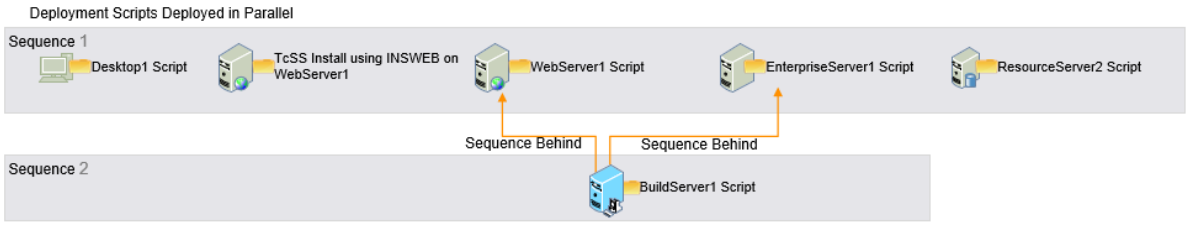
2. **Configure and deploy interactively using Deployment Center Client**

Deployment Center makes configuring a multi-tier deployment with Active Workspace UI Builder is straightforward by offering the option to choose a distributed deployment. Use the following general instructions to configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.

- When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
- On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
- It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
- We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
- To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
- Install "Declarative Artifact Service" (DARSI) microservice by specifying "Instances" value to 1 or more on "Microservice Node" component.
- Follow the instructions given in the Appendix Section 5.2 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.
- The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

- Desktop1 Script
- WebServer1 Script
- EnterpriseServer1 Script

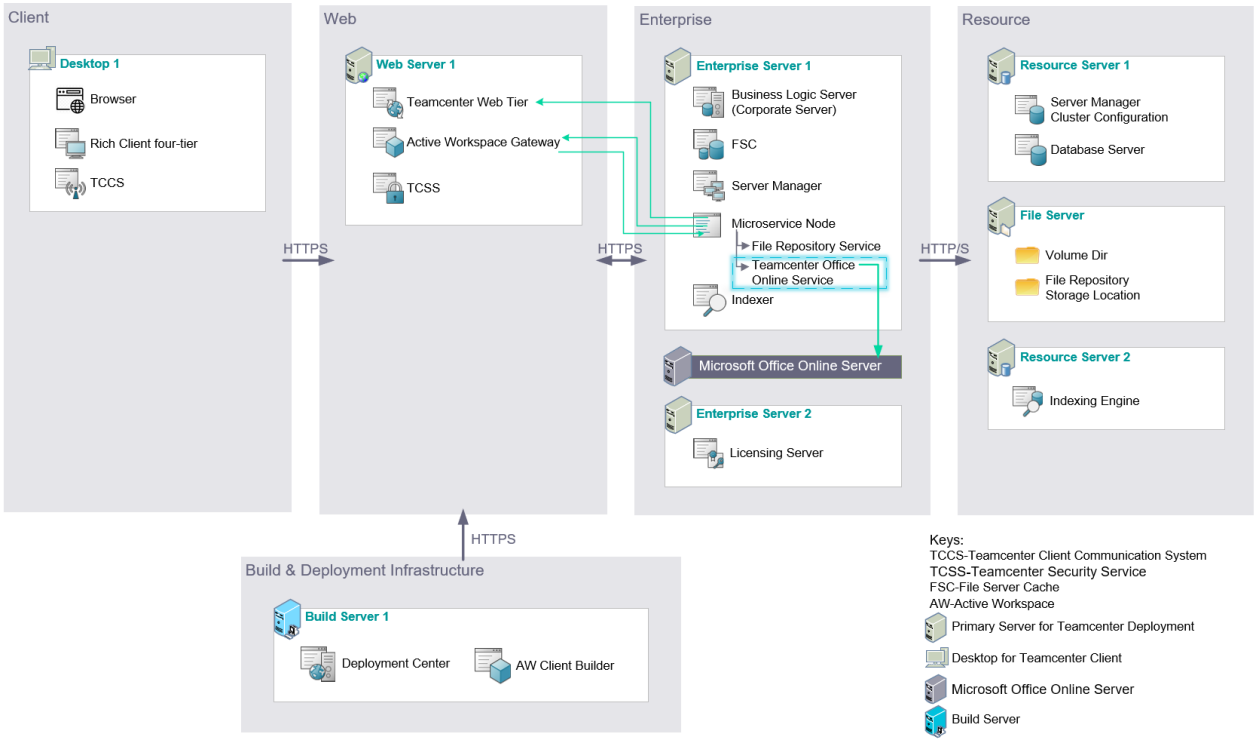
- ResourceServer2 Script

And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.

- BuildServer1 Script must be run after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.

4.9.3 Multi-tier with Teamcenter Office Online

The following diagram shows deployment configuration specific to Teamcenter Office Online Deployment on a multi-tier environment.



Configuration Instructions

This section briefly provides instructions to configure and deploy this "Multi-tier with Teamcenter Office Online" configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.

     \quick_deploy_configurations\wntx64\Teamcenter_RA4.9.3_Multitier_OfficeOnline_Deployment_wntx64.xml" in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.9.3_Multitier_OfficeOnline_Deployment_lnx64.xml" in case of Linux platform.

   - and follow the instruction given in the readme "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

2. **Configure and deploy interactively using Deployment Center Client**

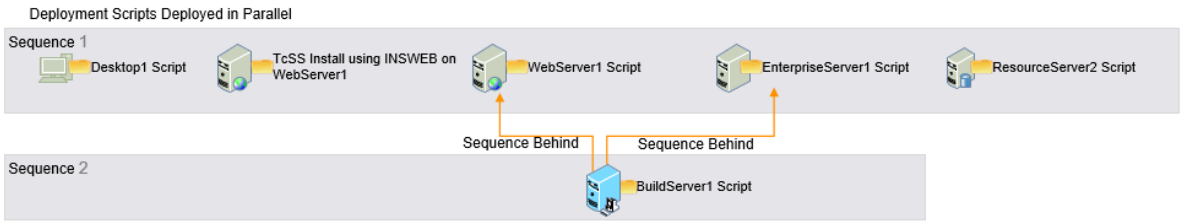
   Deployment Center makes configuring a multi-tier deployment with Teamcenter Office Online is straightforward by offering the option to choose a distributed deployment. Use the following general

instructions to configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.

- When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
- On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
- It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
- We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
- To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
- Install "Teamcenter Office Online Service" microservice by specifying "Instances" value to 1 or more on "Microservice Node" component and configure all office online specific parameter.
- Follow the instructions given in the Appendix Section 5.2 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.
- The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

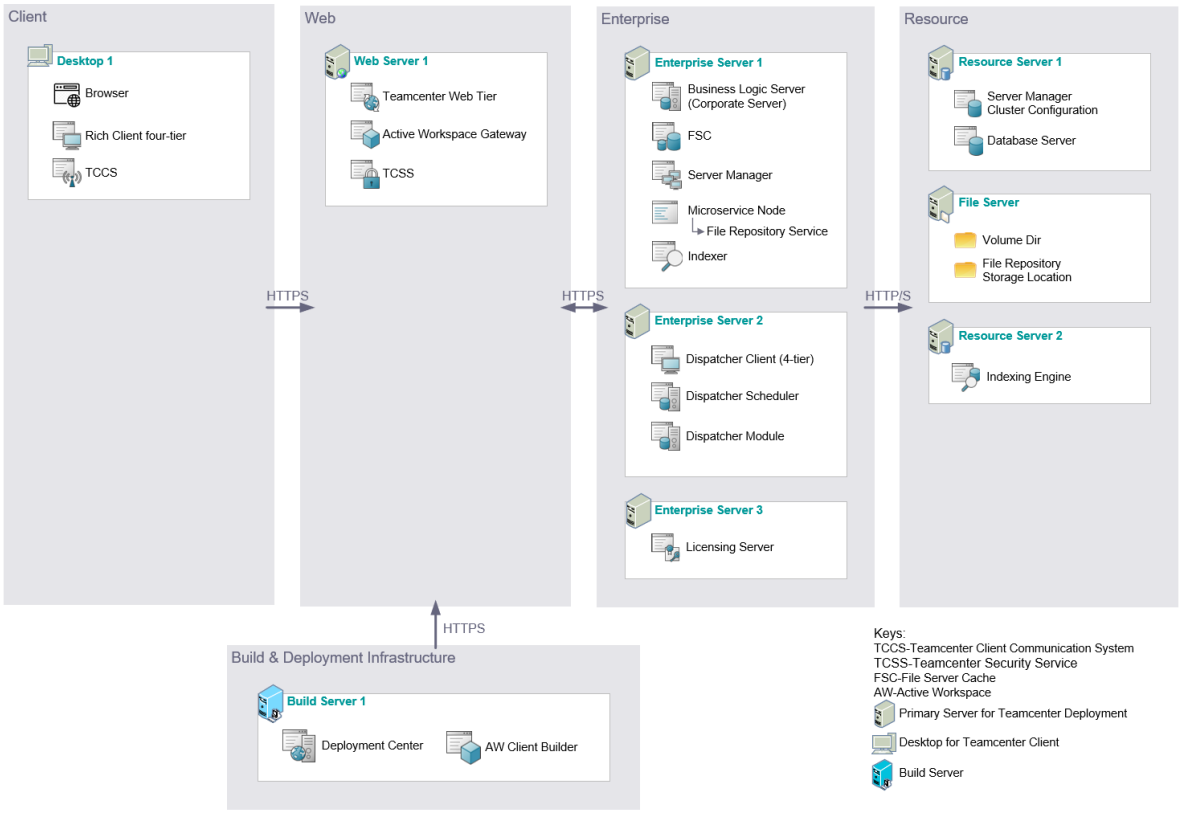
- Desktop1 Script
- WebServer1 Script
- EnterpriseServer1 Script
- ResourceServer2 Script

And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.

- BuildServer1 Script must be run after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.

4.9.4 Multi-tier with Teamcenter Dispatcher

The following diagram shows deployment configuration specific to Teamcenter Dispatcher Deployment on a multi-tier environment.



Configuration Instructions

This section briefly provides instructions to configure and deploy this "Multi-tier with Teamcenter Dispatcher" configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration. \quick_deploy_configurations\wntx64\Teamcenter_RA4.9.4_Multitier_Dispatcher_Deployment_wntx64.xml" in case of Windows platform and \quick_deploy_configurations\lnx64\Teamcenter_RA4.9.4_Multitier_Dispatcher_Deployment_lnx64.xml" in case of Linux platform.
   - and follow the instruction given in the readme "\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt"

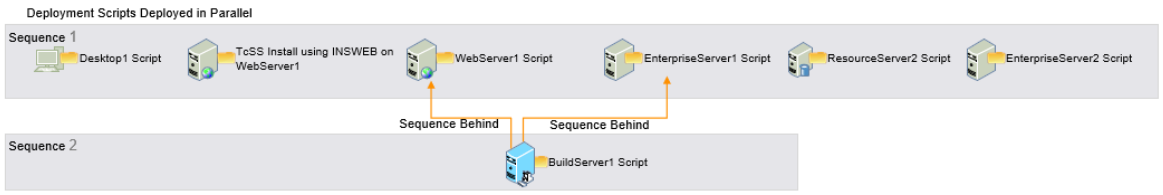
2. **Configure and deploy interactively using Deployment Center Client**
   Deployment Center makes configuring a multi-tier deployment with Dispatcher straightforward by offering the option to choose a distributed deployment. Use the following general instructions to

configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.

- When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
- On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
- It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
- We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
- To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
- Select application "Dispatcher" and desired translators from **"3 Applications"** tab to install Dispatcher & translator.
- On the **4 Components** task, assign same server machine to "Dispatcher Scheduler", "Dispatcher Client (4-tier)" and "Dispatcher Module" to create clustered configuration.
- Follow the instructions given in the Appendix Section 5.2 to configure a Teamcenter environment with SSO using the Teamcenter Security Service.
- The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

- Desktop1 Script
- WebServer1 Script
- ResourceServer2 Script
- EnterpriseServer2 Script
- EnterpriseServer2 Script

And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.

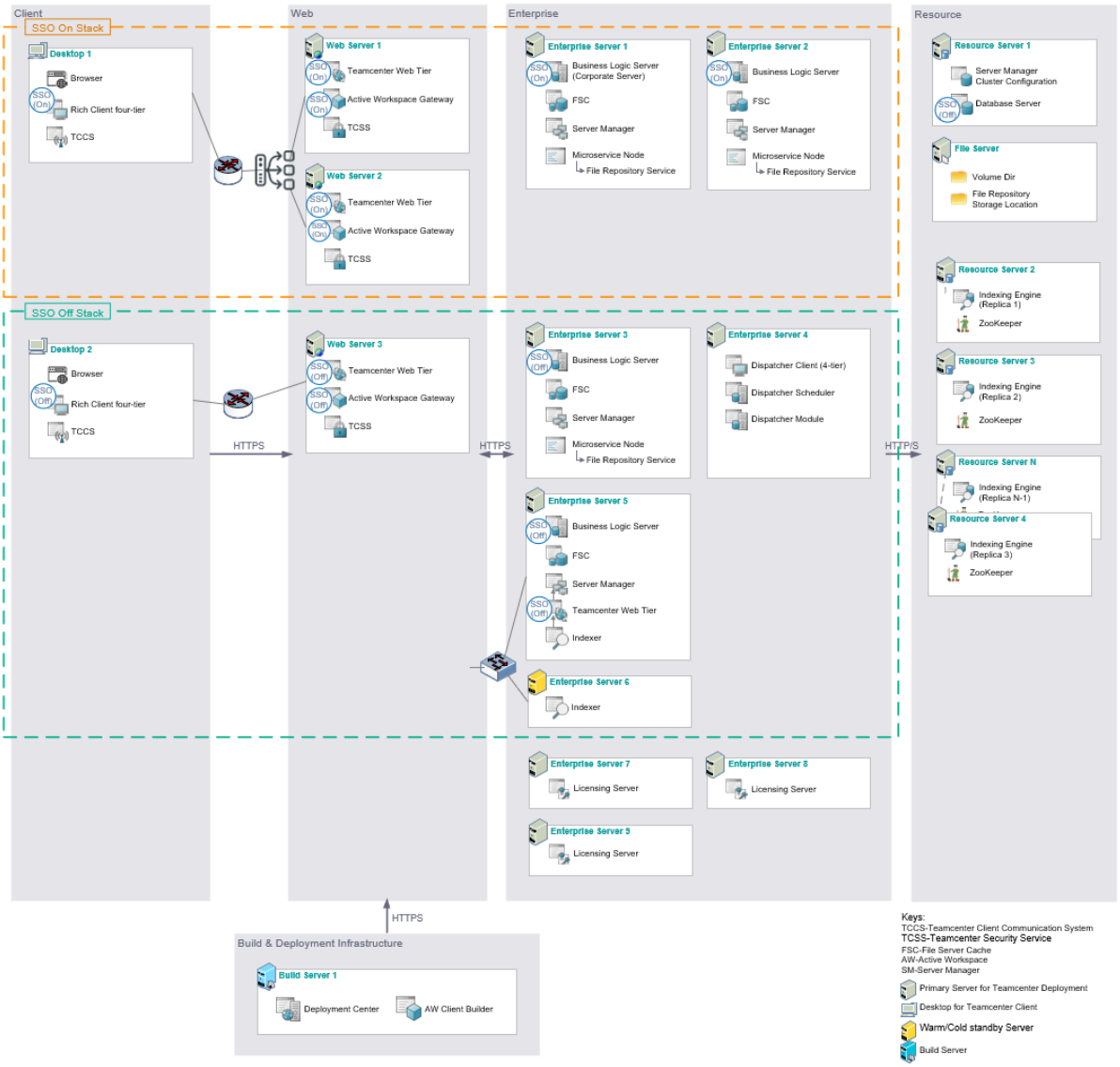
- BuildServer1 Script must be run after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.

4.9.5 Multi-tier with mixed SSO on/SSO off Deployment

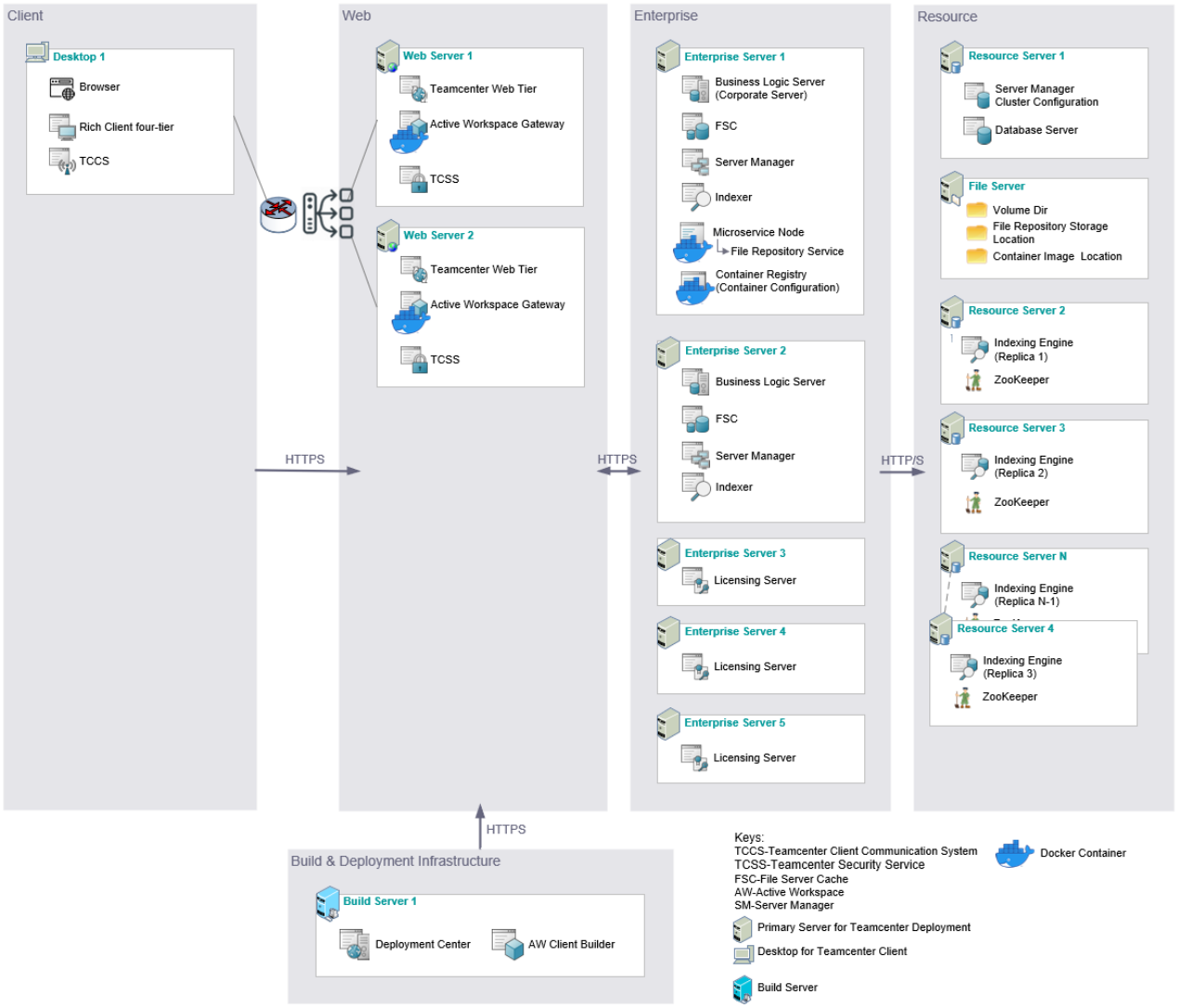
The following diagram shows a deployment configuration example that demonstrates the turning on/off single sign-on (SSO) on portion of the Teamcenter environment. The individual components can be configured to turn on/off single sign-on based on the business requirements.

Note: This diagram serves as an example to show the mixed SSO on/off, but it is not a requirement to have the configuration like this to configure mixed SSO.

Deployment configuration for Windows Platform



Deployment configuration for Linux Platform



Following components are enhanced to support turning single sign-on on/off on individual component.

- Client tier: Rich client four-tier, Rich client two-tier, Teamcenter Rich Client (Shared Disk Deployment)
- Web tier: Teamcenter Web tier, Active Workspace Gateway, Teamcenter Security Service (TCSS)
- Enterprise tier: Corporate Server and Business Logic Server

As shown in the above diagram, portion of the environment (grouped in orange outline as SSO On Stack) is configured to turn on SSO and another portion of the environment (grouped in green outline named as SSO Off Stack) is configured to turn off SSO. In order to have correct SSO behavior, it's critical that SSO supported components participating in a given line of communication across all tiers are either all turned on or all turned off – that is, some cannot be off while others are on.

For example,

- In the “SSO On Stack”, where SSO is configured to turn on
    - the Rich client four-tier (configured to desktop1) that could connect to one of the Teamcenter Web Tier
    - both Teamcenter Web Tier (configured to WebSever1/WebServer2) that could connect to Corporate Server/Business Logic via Server Manager
    - the Corporate Server/Business Logic Server (configured to EnterpriseServer1 /EnterpriseServer2)
- In the “SSO Off Stack”, where SSO is configured to turn off
    - the Rich client four-tier (configured to desktop2) that could connect to one of the Teamcenter Web Tier
    - the Teamcenter Web Tier (configured to WebSever3) that could connect to Corporate Server/Business Logic Server via Server Manager
    - the Corporate Server/Business Logic Server (configured to EnterpriseServer3)

Important note

- If SSO is turned on at database server, then it is required to turn on SSO on the entire environment by configuring to turn on SSO on all the SSO supported components. In other words, SSO cannot be off in components which support SSO.
- If SSO is turned off at database server, then the components can be configured to have mixed SSO (some can be on and some can be off).

Server sizing and performance guidelines

For machine sizing and scalability guidelines, refer to the Teamcenter Hardware Overview document on Support Center for more information.

Configuration Instructions

This section briefly provides instructions to configure and deploy this “Multi-tier with mixed SSO on/SSO Off Deployment” configuration. For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
    - Use the following Quick Deploy Configuration example and readme that is packaged and shipped part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration.

      \quick_deploy_configurations\wntx64\Teamcenter_RA4.9.5_Multitier_MixedSSO_Deployment_wntx64.xml” in case of Windows platform and \quick_deploy_configurations\lnx64\ Teamcenter_RA4.9.5_Multitier_MixedSSO_Deployment_lnx64.xml” in case of Linux platform.

    - and follow the instruction given in the readme “\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt”

2. **Configure and deploy interactively using Deployment Center Client**

Deployment Center makes configuring a multi-tier with mixed single sign-on (SSO) on/off deployment straightforward by offering the option to choose a distributed deployment.

Before configuring environment in the Deployment Center, it is best practice to capture the environment details in the format of Deployment Reference Architecture using the samples provided as part of this release. In the Deployment Reference Architecture, clearly group the components to make stack that will turn on/off the SSO like it is shown in the above sample diagram. This would help to configure the environment correctly so that SSO functionality works as expected.

Once the Deployment Reference Architecture diagram ready as per your business requirements, use the following general instructions to configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.

- When you select Teamcenter and Active Workspace software in Deployment Center, it defaults to single box configuration.
- On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
- It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
- We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
- To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
- Follow the instructions given in the Appendix Section 5.2 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.
- The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

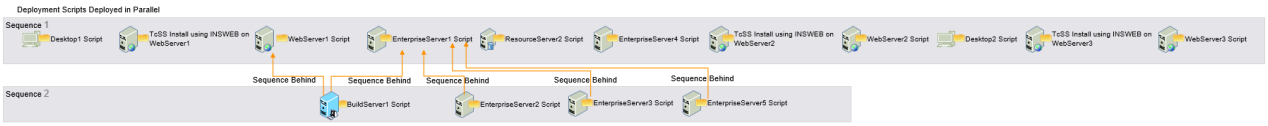
When the environment is configured with Teamcenter Security Service (TcSS) component, the Deployment Center will automatically enable (turns on) the SSO on the entire environment. Follow instructions given below for the individual component, to turn on/off the SSO for the portion of the environment as per the Deployment Reference Architecture diagram,

- Select “Rich client four-tier/Rich client two-tier/Teamcenter Rich Client (Shared Disk Deployment)” component from the Components table on the “4 Components” tab, click on “Show All Parameters” button and turn on/off the SSO by overriding the “Enable Single Sign-On? “ value on “Security Service Settings” section.
- Select “Active Workspace Gateway” component from the Components table on the “4 Components” tab, click on “Show All Parameters” button and turn on/off the SSO by overriding the “Enable Single Sign-On? “ value on “Security Service Settings” section.

- Select “Teamcenter Web Tier (Java EE)” component from the Components table on the “4 Components” tab, click on “Show All Parameters” button and turn on/off the SSO by overriding the “Enable Single Sign-On? “ value on “Security Service Settings” section.
- Select “Corporate Server/ Business Logic Server” component from the Components table on the “4 Components” tab, click on “Show All Parameters” button and turn on/off the SSO by overriding the “Enable Single Sign-On? “ value on “Security Service Settings” section.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

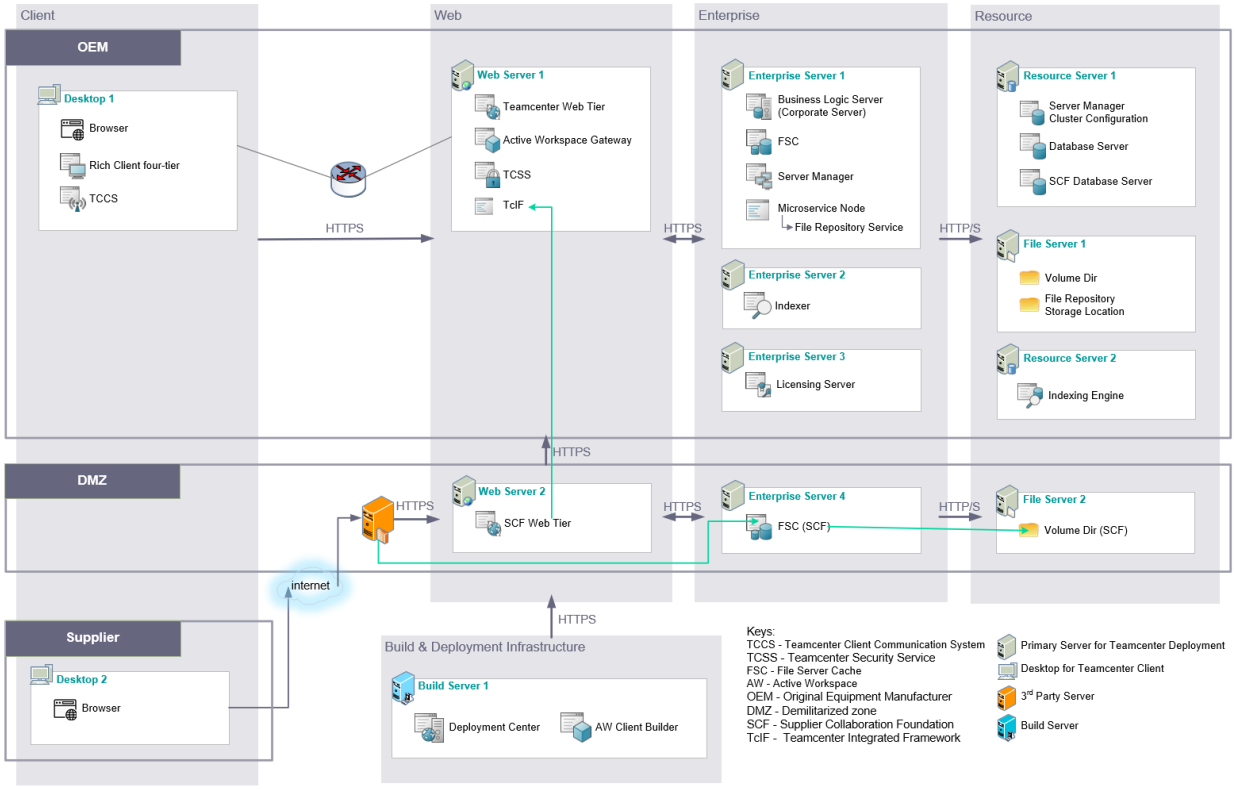
- Desktop1 Script
- WebServer1 Script
- EnterpriseServer1 Script
- ResourceServer2 Script
- EnterpriseServer4 Script
- WebServer2 Script
- Desktop2 Script
- WebServer3 Script

And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.

- BuildServer1 Script must be run only after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.
- EnterpriseServer2 Script, EnterpriseServer3 Script and EnterpriseServer5 Script must be run only after successfully running the scripts EnterpriseServer1 Script

4.9.6 Multi-tier with Supplier Collaboration Foundation

The following diagram shows deployment configuration specific to Supplier Collaboration Foundation Deployment on a multi-tier environment.



Configuration Instructions

This section briefly provides instructions to configure and deploy “Multi-tier with Supplier Collaboration Foundation” configuration. The focus on this configuration is how to deploy FSC and Volume Directory specific to Supplier Collaboration Foundation. Refer to the Supplier Collaboration Foundation deployment guide for more details.
For more detailed instruction, refer to the Deployment Center Help Guide. Follow one of the ways given below to configure and deploy this configuration.

1. **Configure and deploy using Quick Deploy command line utility**
   - Use the following Quick Deploy Configuration example and readme that is packaged and shipped as part of Teamcenter Deployment Reference Architecture downloads to configure and deploy this reference architecture configuration. For this example, it is configured using JBOSS application server and Oracle database server.

     \quick_deploy_configurations\wntx64\Teamcenter_RA4.9.6_Multitier_SCF_Deployment_wntx64.xml” in case of Windows platform and \quick_deploy_configurations\lnx64\ Teamcenter_RA4.9.6_Multitier_SCF_Deployment_lnx64.xml” in case of Linux platform.

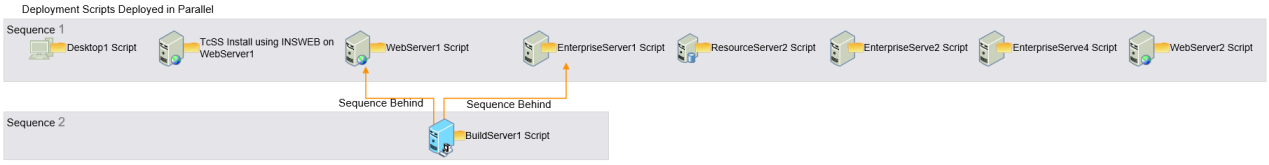
   - Follow the instruction given in the readme “\quick_deploy_configurations\how_to_deploy_using_these_configurations_readme.txt”

2. **Configure and deploy interactively using Deployment Center Client**
   Use the following general instructions to configure a multi-tier distributed deployment. Refer to Teamcenter deployment guide for detailed instructions.

   - When you select Teamcenter, Active Workspace software, Teamcenter Integration Framework, Supplier Collaboration Foundation and Supplier collaboration in Deployment Center, it defaults to single box configuration.
   - On the **2 Options** tab, change the default **Environment Type** from **Single Box** to **Distributed**.
   - It is important to place components on the correct tier. Refer to the diagram in Section 2.1, which indicates in which tiers to place components.
   - We highly recommend you secure the communication between components by choosing the HTTPS communication protocol. Refer to Section 2.2 for information on communication protocol and the default settings to make component configuration appropriately to use HTTPS based communication.
   - To configure components with HTTPS using certificates, follow the manual instructions provided in the Teamcenter help guide.
   - Select application "Supplier collaboration"
   - On the **4 Components** task, assign server machine to "Teamcenter Integration Framework(TCIF)" and "Supplier Collaboration Foundation"
   - Follow the instructions given in the Appendix Section 5.2 to configure a Teamcenter environment with Single Sign-on using the Teamcenter Security Service.
   - The deployment component configuration on both Windows & Linux platforms are similar with the following exception. In Linux, Deployment Center configures Microservice Node and Active Workspace Gateway components to run as container and automatically adds Container Configuration component to get details of the Docker Container Registry. A single container registry should be used to deploy all microservices and Active Workspace Gateway images. In Windows, Deployment Center configures one Microservice Node and Active Workspace Gateway components to run as process.

Deployment Sequence

During deployment, many components can be deployed in parallel, but some components must be deployed in a certain sequence based on the product architecture. Refer to the diagram in Section 2.3 for information on the component sequence dependency. For this configuration, the following diagram shows the sequence of deployment.



The following deployment scripts listed as Sequence 1 can be run parallel in any order,

- Desktop1 Script
- WebServer1 Script
- ResourceServer2 Script

- EnterpriseServer1 Script
- EnterpriseServer2 Script
- EnterpriseServer4 Script
- WebServer2 Script

And following deployment scripts listed as Sequence 2 must be run parallel in any order after Sequence 1 deploy scripts execution completed.

- BuildServer1 Script must be run only after successfully running the scripts WebServer1 Script and EnterpriseServer1 Script.

4.10 Backup and Restore

Teamcenter supports backup and restoration. Backup environments should consist of:

Database:

- Database export files (Oracle: **dmp**. MS SQLServer: **bak**)
- Data files
- Control files
- Redo logs
- Archive logs
- Software installation files / folders

Teamcenter Server:

- All the Volumes (include store and forward)
- TC_DATA Directory
- The TC_ROOT\install directory, which stores configuration data
- The TC_ROOT\bmide directory, which can contain database templates and custom templates under project folders
- All local Business Modeler IDE project folders, including project folders within source control management (SCM) systems
- All custom scripts and processes that are used in the environment
- Task Scheduler/Cron related items
- Software installation files and folders

Web Application server:

- Deployed files
- Modified configuration files
- Scripts developed to start, stop, status, monitor
- Software installation files and folders

Local and Remote Backup

- **Local Backup**: Data can be backed up to another hard drive, other media, or a shared drive either manually or at specified intervals. With this setup, all the data is within the same premises. Resorting data locally is easy, but the backup data could be at risk due to some unforeseen events like natural disaster.
- **Remote Backup**: Data is sent to a remote center at specified intervals. In the event of some unforeseen events like natural disaster, the data is secured at remote location / or even on the cloud and could be restored when the Internet connection is restored.

Types of Backup:

- A full backup is the most complete type of backup. It is more time-consuming and requires more storage space than other backup options.
- An incremental backup only backs up files that have been changed or newly created since the last incremental backup. This is faster than a full backup and requires less storage space. However, in order to completely restore all the files, you will need to have all incremental backups available. To find a specific file, it may need to search through several incremental backups.

- A differential backup also backs up a subset of the data, like an incremental backup. But a differential backup only backs up the files that have been changed or newly created since the last full backup.

Teamcenter's integrated backup and recovery feature facilitates third-party backup systems to perform online backups, allowing Teamcenter to operate continually. We recommend you choose a third-party software provider that is reliable, stable, and secure. You can also secure data by encrypting it before it is transmitted to the remote site. We also recommend that you test your backups and practice restoration procedures.

5 Appendix

5.1 Tier Architecture

Teamcenter provides a flexible four-tier architecture that allows small businesses to deploy as effectively as the largest global enterprises. Each tier plays a specific role in the Teamcenter deployment and thus requires difference resource needs.

Following is the information about each tier and its role

- Client tier: Hosts client applications and secure file caches, and provides user interface input and output processing
- Web tier: Provides web-enabled access to Teamcenter services and enforces network security. The web tier communicates with the enterprise tier.
- Enterprise (business logic) tier: Hosts business logic, applies security rules, and serves dynamic content to clients. This tier is computationally intensive and requires a large amount of memory (RAM) for efficient performance.
- Resource tier: Stores persistent metadata in tables and persistent bulk data as files. This tier is the second most resource-intensive because it includes the database server. Unlike the business logic tier, the database depends heavily on an efficient input/output (I/O) configuration rather than large amounts of memory.

5.2 Instructions to configure a Teamcenter environment with single sign-on

Follow the instructions given in the Security Services Installation/Customization guide to configure and deploy Teamcenter Security Service using INSWEB. On successful installation of the Teamcenter Security Service, follow the instructions given below in Deployment Center to configure the Teamcenter environment with single sign-on using Teamcenter Security Service.

- Select an environment that need to be configured with single sign-on authentication.
- From **4 Components** task, add **Teamcenter Security Services (TcSS)** component from the available optional component list.
- Configure this component correctly by specifying appropriate values from actual installation of Teamcenter Security Service using INSWEB.
- Once the component is configured correctly and saved, Deployment Center automatically configures Client, Server, and other components to enable the single sign-on authentication. During deployment, these settings will be used to create the respective component configurations enabled with single sign-on.

5.3 Instructions to enable single sign-on/Teamcenter Security Service behind a firewall

- Follow the instructions in the *Security Services Installation/Customization* guide to configure and deploy Teamcenter Security Service behind a firewall using INSWEB. On successful installation of Teamcenter Security Services.

- To enable the Teamcenter Security Service behind a firewall, from the **4 Components** task:
    - Select **Teamcenter Client Communication System** component configured for the client machine and check the **Enable Client Communication System Configuration** checkbox on the Advance Parameter panel.
    - Configure the **Forward Proxy Settings** on the Teamcenter Client Communication System component appropriately for the forward proxy server. Choose **Configure settings manually** to add and configure the **Forward Proxy Server** installed as a third-party solution:
        - Add the **Forward Proxy Server** component from the available optional component list and configure this component correctly by specifying appropriate values based on your actual installation of the forward proxy server.
        - Once the **Forward Proxy Server** component is configured and saved, Deployment Center automatically configures **Forward Proxy Settings** on the Teamcenter Client Communication System component to reference **Forward Proxy Server**.
    - Configure the **Reverse Proxy Settings**, **Kerberos Authentication Settings** and **Secure Socket Layer (SSL) Settings** options on the **Teamcenter Client Communication System** component appropriately, based on the reverse proxy server installed, and its third-party authentication configuration.

6 Customer Support

6.1 Installation assistance

For additional installation assistance, or to report any problems, contact Customer Support.

Website:
https://support.sw.siemens.com/

Phone:
United States and Canada: 800-955-0000 or 714-952-5444

Outside the United States and Canada: Contact your local support office.

About Siemens Digital Industries Software

Siemens Digital Industries Software is a leading global provider of product life cycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens Digital Industries Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens Digital Industries Software products and services, visit **www.siemens.com/plm**.