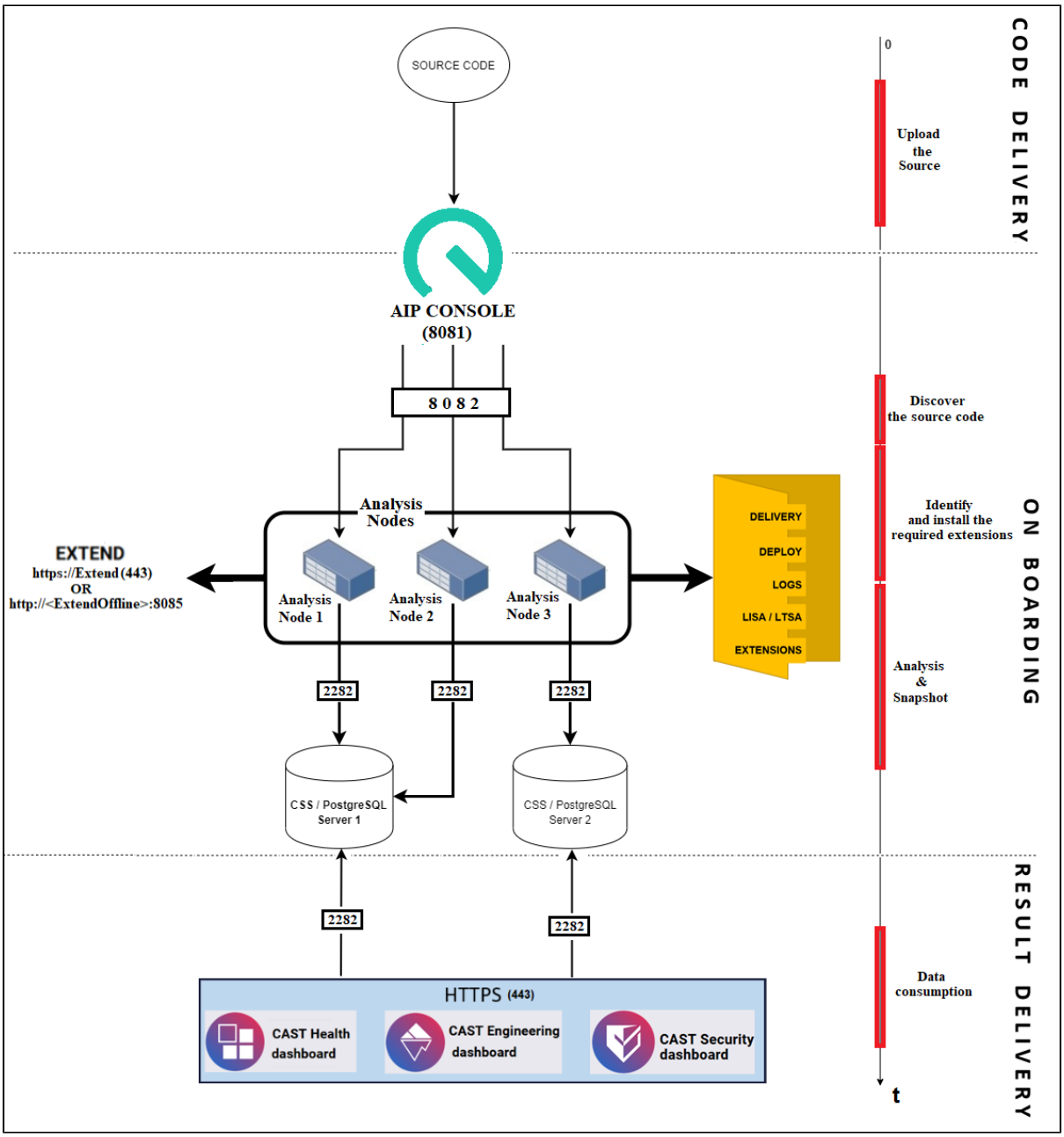
Deployment considerations

Summary: information to help you plan your **CAST AIP for Dashboards** installation.

Typical communication between CAST components

The following diagram shows the interaction between **AIP Console** and the **main CAST Components**. It also shows the flow from the source code delivery to the snapshot and Dashboard data consumption.

Click to enlarge



*Analysis Node = Server on which the **AIP Node package** and **AIP Core** are installed.

Required components

Package/Component	Description
AIP Core	**AIP Core** is the "backbone" analysis and snapshot engine. • This component can be installed on **Windows** servers only. • Can only be installed **once per instance of Windows**.

AIP Node package (back end)	The **AIP Node "back end" package** contains the RESTful APIs that interact with CAST AIP on the same host server and the remote CAST Storage Service/PostgreSQL and AIP Console instances. This package: • should be installed on **all servers** where **AIP Core is already installed**. • is provided in the AIP Console ZIP file, as an installable .JAR file together with the AIP Console (front end). • can be installed on **Windows** servers only. • Can only be installed **once per instance of Windows**.
CAST Storage Service / PostgreSQL	This component provides the storage (i.e. database schemas) for the analysis/snapshot result data produced by AIP Console. It can be installed: • on **Windows** (use the prepackaged CAST Storage Service) or **Linux servers** (install PostgreSQL) • multiple times, i.e. you can use multiple CAST Storage Service/PostgreSQL instances • **BEFORE** you install the AIP Console and AIP Nodes packages. CAST highly recommends the use of PostgreSQL on a Linux instance as this consistently gives the best performance.
AIP Console package / Dashboards (front end)	The **AIP Console "front end" package** connects and manages all the AIP Node(s). Users will connect (using a browser) directly to this package to manage the analyses of their Applications. This package: • should only be installed once. • can be installed on **Windows** or **Linux servers**. • is provided in the AIP Console ZIP file as an installable .JAR file, together with the AIP Node (back end). • contains the Health and Engineering Dashboards. • From **v. 1.16**, AIP Console can manage multiple **AIP Nodes** using **different releases of AIP Core**. In previous releases, the **same release of AIP Core** must be used across all AIP Nodes managed in the same AIP Console installation.

Optional components

Package /Component	Description

CAST RESTAPI (Dashboard back end)	This components is **only required** if your organization is using the Health and Engineering Dashboards embedded in AIP Console. If you decide to use **standalone dashboards**, this component is not required.
	This component acts as an **intermediary** between AIP Console/Dashboards and the CAST Storage Service/PostgreSQL instances where the analysis data is stored. It can be installed:
	• on **Windows** or **Linux servers**
	• is provided in the AIP Console ZIP file as:
	  • a **bootable ZIP file** not requiring Apache Tomcat **(AIP Console 1.25)**
	  • a **.WAR file** requiring Apache Tomcat **(AIP Console 1.24)**

CAST Extend Offline or **CAST Extend local server**	These components are **only required** if (due to security concerns) your organization/AIP Console cannot interact over the public internet with **CAST's Extend system for Extension management**. The components provide CAST's Extend system as an "offline" on-premises component:
	• should only be installed once
	• can be installed **only** on **Windows servers**
	• should ideally be installed **BEFORE** you install the AIP Console and AIP Nodes packages. This is so that when running the initial **start-up wizard for AIP Console**, you already have all the details that are required.
	• All AIP Nodes require access to CAST Extend Offline/Proxy.
	• If the AIP Node accessing CAST Extend Offline/Proxy is configured to pass all outgoing connections through a proxy (via the Windows proxy settings or via the AIP Console settings), then you may need to whitelist the IP address/host name of the server running CAST Extend Offline/Proxy in order to route connections correctly.
	• CAST Extend Proxy can be configured in offline or online mode (CAST Extend Offline is always offline):
	  • When in offline mode, functionality is the same as CAST Extend Offline and extensions must be added manually to CAST Extend Proxy.
	  • When in online mode, CAST Extend Proxy is configured to connect with CAST's Extend system for Extension management over the public internet.

Storage considerations

As shown in the above diagram, there are five main storage folders (marked in yellow). All AIP nodes must have access to these folders:

Delivery	Used for storing **successive and compressed versions of an application's source code** produced during the source code delivery phase. This folder contains all the **versions** of the source that have been delivered. This folder is usually the **largest** of all storage folders.
Deploy	Used for storing an application's **source code in uncompressed format** - this is the code that is analyzed. When the delivery is **accepted and imported**, the **deploy folder** is cleaned and the current version of the source code is deployed in the **deploy folder**.
Logs	Used for storing all logs produced by the AIP Node with regard to code delivery/analysis/snapshot activities. One sub-folder folder will be created per Application onboarded in AIP Console. The log files will contain the path and the name of the source file. For some warning messages (e.g.: syntax error), the path of the source code involved in the warning is shown in the log.
LISA / LTSA	Location to store **temporary files** generated during the analysis/snapshot process on each AIP Node: • **Large Intermediate Storage Area (LISA)** - cleaned on completion of each analysis. • **Large Temporary Storage Area (LTSA)** - is permanent and contains preprocessed source code files, archive files for analysis etc. Note that some analyzers/extensions, need to preprocess the source code before performing the analysis. In this case, the source code that will be analyzed is stored in the **LISA** folder, not in the **Deploy** folder.
Extensions	Used for storing AIP extensions.

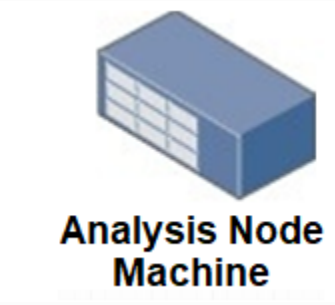
When a **snapshot** is generated, the analyzed source code files are uploaded into the **Analysis schema (local)** and the **Dashboard schema (central)**.

Storage folder distribution strategy

This section explains the different ways that can be used to distribute the required storage folders:

- Local distribution only
- Local and network distribution
- Network distribution only

We also describe the relationship between performance and scalability depending on the way you choose to deploy the storage folders. In addition, we suggest some required disk space recommendations (minimum recommendations and advised configuration for a large number of applications) depending on the chosen storage folder distribution strategy.

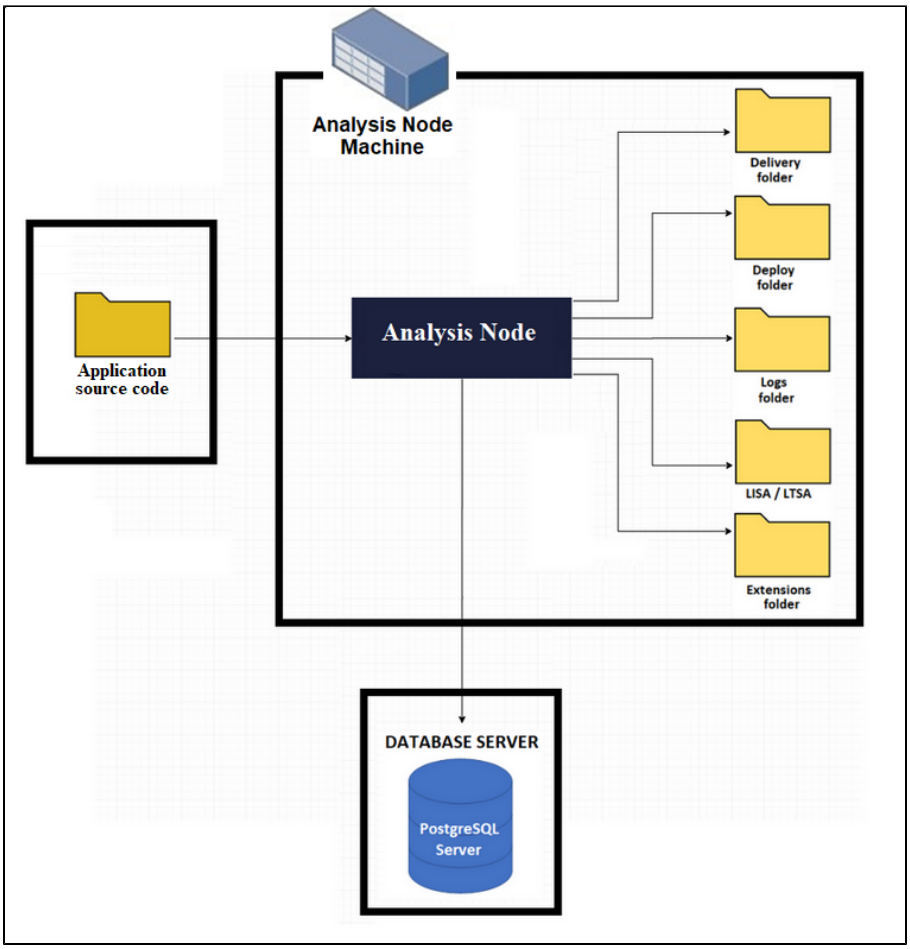


- This icon depicts the server on which the back-end AIP Node package and AIP Core are installed.
- CAST Storage Service / PostgreSQL can be installed on the AIP Node or on a separate and dedicated server.

Local distribution: better performance / lower scalability

All the storage folders are on the local server - i.e. the AIP Node:

Click to enlarge



The limiting factor for this deployment is that all these folders **can grow very quickly during an analysis**, therefore, they can potentially overload the local disks on the server.

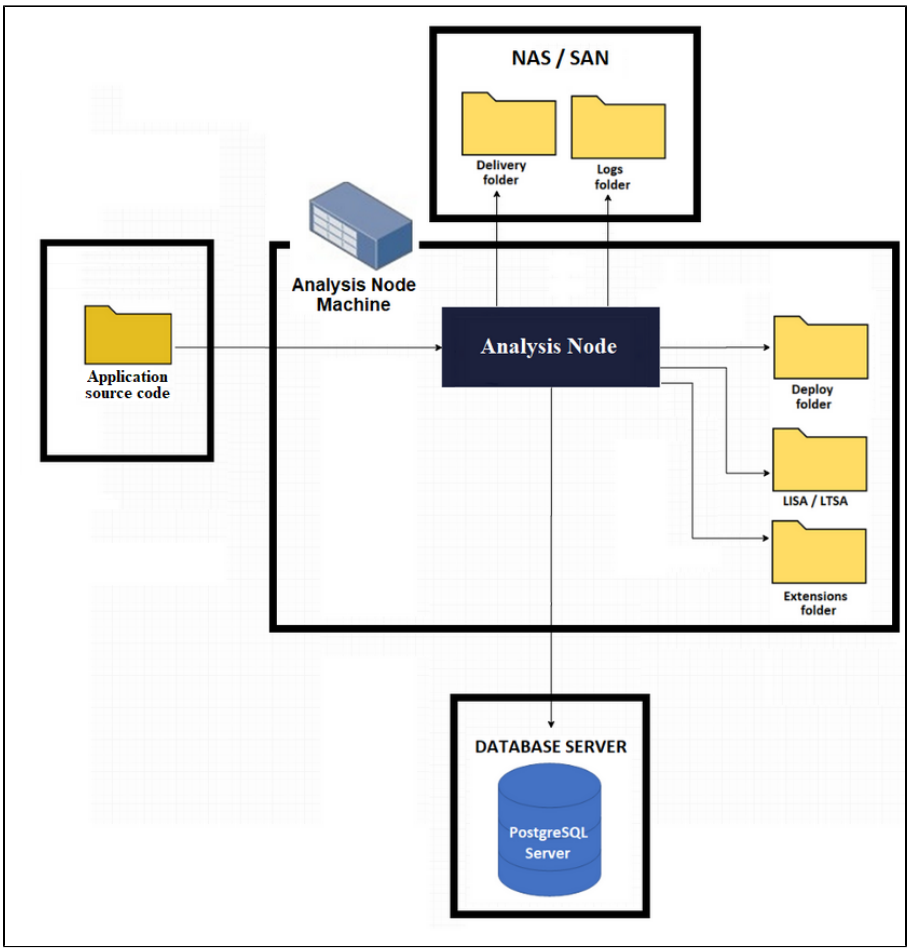
Disk space recommendations

Minimum recommendation	Local storage (all folders)	256GB
Run multiple applications in parallel (minimum)	Local storage (all folders)	1TB

Balance between storage and performance: medium performance / medium scalability

The **Delivery** and the **Logs** folders are located on a shared network drive. Other folders are located on the local server - i.e. the AIP Node:

Click to enlarge



The aim of using a shared network folder for the **Delivery** and **Logs** folders is to accommodate the potentially large amount of storage that is required. The size of these two folders can grow very quickly during an analysis (in particular the Delivery folder which stores multiple versions of a given Application's source code).

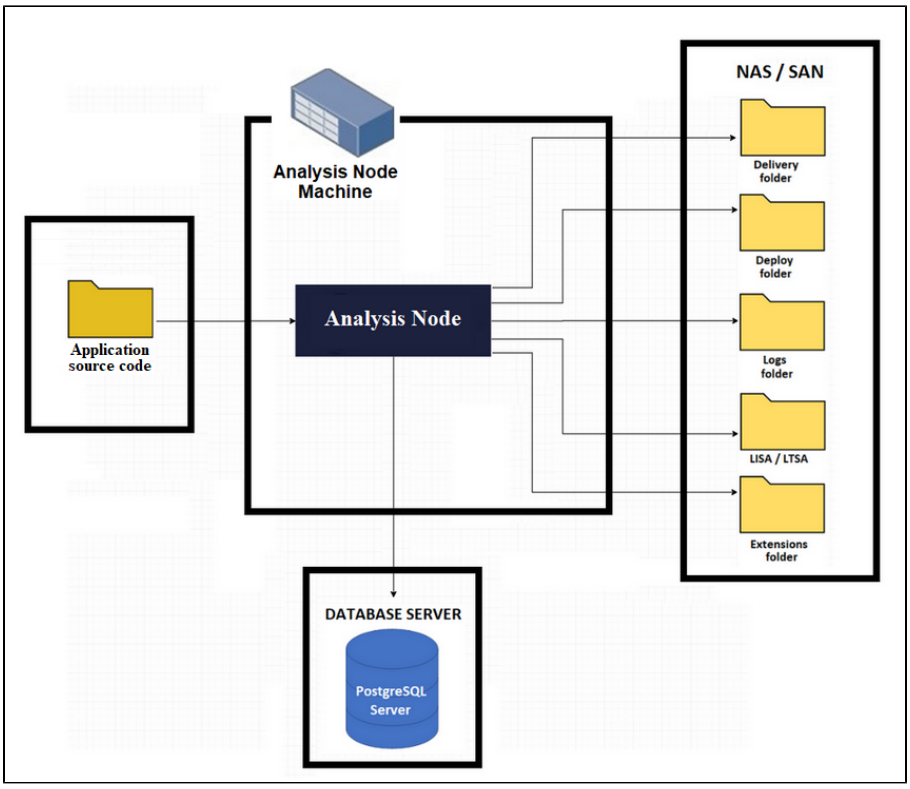
This configuration reduces the risk of overloading the local server's disk space, however, performance during analysis will be slightly reduced since the log file is located on a shared network drive, and an analyzer will require slightly more time to write to it during an analysis.

Disk space recommendations

Minimum recommendation	Local storage (Deploy, LISA and LTSA and Extensions folders)	86GB
	Network storage (Delivery and Log folders)	170GB
Run multiple applications in parallel (minimum)	Local storage (Deploy, LISA and LTSA and Extensions folders)	300GB
	Network storage (Delivery and Log folders)	700GB

Network distribution: lower performance / better scalability

All the storage folders are located on a shared network drive:



This configuration will never overload the local disk space, however, performance during analysis is slower compared to the first deployment scenario for the following reasons:

1. The time required to load the source code in to memory from a network drive is inevitably longer.
2. Slower log file updates
3. Temporary files will be generated more slowly.

Disk space recommendations

Minimum recommendation	Network storage (All folders)	256GB
Run multiple applications in parallel (minimum)	Network storage (All folders)	1TB

Note that to analyze .NET applications with assemblies stored in a remote network location, you will need to alter the machine configuration by proceeding as described in the .NET Analyzer in the section **Prerequisites**.

RAM considerations for AIP Console (front-end) and AIP Nodes

See **Configuring RAM for AIP Console front-end and AIP Nodes** for more information.

Windows Services user account considerations - Microsoft Windows only

Both the AIP Console front-end and the AIP Node back-ends are run as Windows Services on the relevant Windows machine. CAST highly recommends that the **Local System** account is **not used** to run these Windows Service. This is particularly true if:

- you need to use a **proxy service** to access specific resources such as CAST Extend (see **Complete start-up wizard** or **Administration Center - Settings - Proxy**)
- you are using **shared network resources** to store data such as Delivery and Deploy items (see **Configure AIP Node storage folder locations - optional**)

In both these situations, the user running the Windows Service will be used to access the proxy/shared network resources.

Instead, CAST recommends using the login credentials that match the log in used to install AIP Console/AIP Node/AIP Core/set system proxy settings etc. - for example, this could be a specific "**service account**" that is created specifically for installing and running AIP Console/AIP Nodes/AIP Core/setting system proxy settings. This service account would also therefore have access to the shared network resources and would be able to use the system proxy settings.

Proxy considerations

If your organization requires the use of a proxy, please take the following into account:

Extension Downloader limitation

At the current time, the **Extension Downloader** (a tool present on each AIP Node which is used by AIP Console to download extensions) cannot be configured to obey a **manual proxy configuration** defined in AIP Console. Instead, if your organization uses a proxy, CAST recommends that:

- you define the required proxy configuration at **system level** (i.e. operating system level) on **all AIP Nodes**
- define a **manual proxy configuration** using the settings described below - this ensures that everything else will connect through the proxy

Windows Services

AIP Console and the AIP Nodes packages are configured to run through Windows Services, therefore it is important to ensure that the user login configured to run the Windows Services has permission to access any proxy that you define. If the user running the Windows Services cannot access the proxy, then AIP Console/AIP Nodes will not be able to access the required resources. See the sections regarding the configuration of the Windows Services in:

- **AIP Console package - front-end installation**
- **AIP Node package - back-end installation**

Which deployment architecture should you use?

Deployment scenario	Component	Server Type	OS	Recommended deployment architecture
Multi-node (Large /Enterprise)	**AIP Core** **AIP Node (back end)**	Physical / Virtual	Microsoft Windows	**AIP Core/AIP Node package** • installation of **AIP Core** once per dedicated **Windows server**. • installation of the **AIP Node** package once on all Windows servers where **AIP Core** is already installed. • together both packages form an "**AIP Node**": multiple AIP Nodes can be managed through AIP Console.
	AIP Console (front end)	Physical / Virtual	Microsoft Windows or Linux	**AIP Console "front end" package** • installation of the **AIP Console** package on a dedicated **Windows** or **Linux server** to manage all the AIP Nodes.
	CAST Storage Service (CSS) / PostgreSQL	Physical / Virtual	Microsoft Windows or Linux	**CAST Storage Service / PostgreSQL** Details of the **CAST Storage Service/PostgreSQL** instance will be required during the installation of **each AIP Node package**: all that is required is that the AIP Console installer can access all required CAST Storage Services/PostgreSQL. How you choose to deploy the CAST Storage Service/PostgreSQL generally depends on the types of Application you are onboarding using the AIP Console, however, CAST recommends the following as a **minimum**: • 1 x CSS/PostgreSQL (on a dedicated Windows (CSS) or Linux (PostgreSQL) server) **for all small and average sized Applications (i.e. multiple AIP Nodes per CSS/PostgreSQL instance)** • 1 x CSS/PostgreSQL (on a dedicated Windows (CSS) or Linux (PostgreSQL) server) **per Application for all large Applications (i.e. one dedicated AIP Node + dedicated CSS/PostgreSQL instance per large Application)** • 1 x CSS/PostgreSQL (on a dedicated Windows (CSS) or Linux (PostgreSQL) server) **for the MEASUREMENT schema**

	Dashboard RESTAPI	Physical / Virtual	Microsoft Windows or Linux	**Dashboard RESTAPI**
				To deploy CAST dashboards and leverage the auto-update capabilities provided by the AIP Console, a **deployed CAST RESTAPI** must be available. CAST recommends using:
				• 1 x dedicated **Windows** or **Linux server** to host the **CAST RESTAPI:**
				  • for new releases of AIP Console (1.25) a bootable ZIP is provided which does not require Apache Tomcat.
				  • for older releases of AIP Console (1.24) Apache Tomcat must also be deployed to host the CAST RESTAPI.
				You can also re-use another of your servers to host this component.

	CAST Extend Offline or CAST Extend local server	Physical / Virtual	Microsoft Windows	**CAST Extend Offline or CAST Extend local server** If you need to deploy the **CAST Extend Offline** or **CAST Extend local server**, details of this service will be required during the **initial start up of AIP Console**: all that is required is that the wizard can access CAST Extend Offline/Proxy. CAST recommends using either: • a dedicated **Windows server** to host the service • re-use one of the existing servers dedicated to the CAST RESTAPI or AIP Node, providing that this is a **Windows server** • If the AIP Node accessing the CAST Extend Service is configured to pass all outgoing connections through a proxy (via the Windows proxy settings or via the AIP Console settings), then you may need to whitelist the IP address/host name of the server running CAST Extend Service in order to route connections correctly.

Single server (Small/Simple)	**AIP Core**
	AIP Node (back end)
	AIP Console (front end)
	CAST Storage Service (CSS) / PostgreSQL*
	Dashboard RESTAPI
	CAST Extend Offline or CAST Extend local server
		Physical / Virtual	Microsoft Windows	If you have one single **AIP Core installation** that you would like to manage with the AIP Console, choose this option, which involves the installation of **all packages/components** on one single **Windows server** (i.e. the server on which CAST AIP is already installed).
				
				Note that:
				
				• Although it is possible to run all required packages and components from one single Windows server (for example in very small deployments or for testing purposes), CAST does not recommend this approach for "production" environments.
				• Details of the **CAST Storage Service/PostgreSQL** will be required during the installation: all that is required is that the AIP Console installer can access it.
				• * The CAST Storage Service/PostgreSQL instance does not necessarily need to be installed on the same server on which an **AIP Node** is installed:
				  • It is required that the AIP Console installer can access it.
				  • For large applications, it is strongly advised that a **PostgreSQL Server** is installed on a **Linux server**.
				• If you want to deploy CAST dashboards and leverage the auto-update capabilities provided by the AIP Console, then a deployed **CAST RESTAPI** file must be available. In a single-server scenario, CAST recommends using the same server on which the CAST AIP Console / CAST AIP is installed.
				• Details of **CAST Extend Offline/Proxy** (should you wish to deploy it) will will be required during the installation: all that is required is that the AIP Console installer can access it.

Type of Applications being onboarded

The **method** and **hardware** chosen to deploy CAST components and packages depends on the **type of the Application(s)** you intend to onboard with AIP Console. Various factors related to the Application(s) you are analyzing must be considered:

- Number of lines of code (LOC) in the Application (in general, CAST considers an Application with **over 3 million lines of code** to be "large", however, this figure is **not set in stone** and smaller Applications may require dedicated resources).
- Number of objects after analysis
- Number of violations after analysis
- Technologies used
- Number of users accessing results

When onboarding "large" applications you should carefully consider how you deploy the required components and packages - particular attention should be paid to:

- **AIP Core/AIP Node** - dedicate one server for each large Application - or if this is not possible, run analyses/snapshots of one single large application one at a time
- **CAST Storage Service / PostgreSQL** - dedicate one server and CAST Storage Service/PostgreSQL instance for each large Application - use Linux/PostgreSQL to improve performance

What about when transitioning from legacy CAST Management Studio to AIP Console?

If you are an **existing CAST customer** already using "legacy" CAST Management Studio and you are planning a transition to AIP Console to **onboard new Applications**, then you can reuse your existing deployments of:

- **AIP Core** already installed - install the **AIP Node package** on each to enable it for use with AIP Console
- **CAST Storage Service/PostgreSQL** - declare this server(s) when installing the AIP Node package(s)
- **CAST Dashboards** - with regard to dashboards, the choice is yours:
  - there are advantages of using **existing deployed dashboards** - no new installation/configuration required, can be fully customized as required (customization not currently possible with dashboards embedded in the AIP Console package)
  - there are advantages of using **dashboards embedded in the AIP Console package** - seamless login from with Console credentials, fully embedded in the Console process

For those wishing to transition existing Applications to AIP Console, a **specific process is in place**, but you can re-use your existing deployments as above.