Once you get Tomcat up and running on your server, the next step is configuring its basic settings. Your initial configuration process will consist of two tasks, which are explained in detail in this article. The first is [editing Tomcat's XML configuration files](https://www.mulesoft.com/tcat/tomcat-configuration#xml_config), and the second is [defining appropriate environment variables](https://www.mulesoft.com/tcat/tomcat-configuration#env_variables).

XML Configuration Files

The two most important configuration files to get Tomcat up and running are called server.xml and web.xml. By default, these files are located at TOMCAT-HOME/conf/server.xml and TOMCAT-HOME/conf/web.xml, respectively.

### SERVER.XML

The server.xml file is Tomcat's main configuration file, and is responsible for specifying Tomcat's initial configuration on startup as well as defining the way and order in which Tomcat boots and builds. The elements of the server.xml file belong to five basic categories - Top Level Elements, Connectors, Containers, Nested Components, and Global Settings. All of the elements within these categories have many attributes that can be used to fine-tune their functionality. Most often, if you need to make any major changes to your Tomcat installation, such as specifying application port numbers, server.xml is the file to edit.

### Top Level Elements

### Server

This element defines a single Tomcat server, and contains the Logger and ContextManager configuration elements. Additionally, the Server element supports the "port", "shutdown", and "className" attributes.

The port attribute is used to specify which port Tomcat should listen to for shutdown commands. The shutdown attribute defines the command string to be listened for on the specified port to trigger a shutdown. The className attribute specifies which Java class implementation should be used.

### Service

This element, which can be nested inside a Server element, is used to contain one or multiple Connector components that share the same Engine component. The main function of this component is to define these components as a single service. The name of the service that will appear in logs is specified using the Service element's "name" attribute.

### Connectors

By nesting one [Connector](https://www.mulesoft.com/tomcat-connector) (or multiple Connectors) within a Service tag, you allow Catalina to forward requests from these ports to a single Engine component for processing. Tomcat allows you to define both HTTP and AJP connectors.

### HTTP Connector

This element represents an HTTP/1.1 Connector, and provides Catalina with stand-alone web server functionality. This means that in addition to executing servlets and [JSP](https://www.mulesoft.com/tomcat-jsp) pages, Catalina is able to listen to specific TCP ports for requests. Each Connector you define represents a single TCP port Catalina should listen to for HTTP requests. When configuring your HTTP connectors, pay close attention to the "minSpareThreads", "maxThreads", and "acceptCount" attributes. The "maxThreads" attribute is of particular importance. This attribute controls the maximum number of threads that can be created to handle requests exceeding the number of available threads. Setting this value too low will cause requests to stack inside the server socket, which will begin refusing connections once it is full. Comprehensive testing will help you avoid this problem.

### AJP Connector

This element represents a connector that is able to communicate with the AJP protocol. The main role of this element is to help Tomcat integrate with an installation of Apache. The most common reason why you would want this functionality is if you plan to use Apache to serve static content in front of Tomcat. This technique is intended to free up more power for dynamic page generation and load balancing, so if fast [performance](https://www.mulesoft.com/tomcat-performance) is a concern for your application, this is something to consider. AJP Connectors can also be used to expose Apache's [SSL](https://www.mulesoft.com/tomcat-ssl) processing functionality to Tomcat.

https://www.mulesoft.com/tcat/tomcat-configuration