You can get on-board this exciting career path with a little help with this [**tutorial on Amazon Web Services certified solutions**](https://www.udemy.com/aws-certified-solutions-architect/?tc=blog.webservicesinterviewquestions&utm_source=blog&utm_medium=udemyads&utm_content=post28220&utm_campaign=content-marketing-blog&xref=blog) and this[**cloud computing guide to AWS**](https://www.udemy.com/cloud-computing-with-amazon-web-services-part-1/?tc=blog.webservicesinterviewquestions&couponCode=half-off-for-blog&utm_source=blog&utm_medium=udemyads&utm_content=post28220&utm_campaign=content-marketing-blog&xref=blog).

First off, what are web services? The simplest answer would be that a web service is any piece of software available over the Internet that uses a standardized XML messaging system. To drill down, XML is a language used to encode all communications to a Web service. Because all communication is in XML, Web services are not limited to any one operating system or programming language. For instance, Java can talk with Per, Windows applications can talk with Unix applications, and so forth.  
Beyond this basic definition, a Web service may further be defined as such:

1. A web service can have a public interface, defined in a common XML grammar. The interface describes all the methods available to clients and specifies the signature for each method. The interface definition is currently created using the Web Service Description Language (WSDL).

2. If you create a Web service, there should be some relatively simple mechanism for you to publish it. This would be the UDDI, Universal Description, Discovery, and Integration mechanism that allows the creators to house and service its public interface. Web services currently run a diversified range of production from news syndication and stock-market data to weather reports and package-tracking systems.

Remember these definitions as interviewers are likely to ask how you classify web services. Now let’s move on to frequently asked questions during web services interviews.

**What are the components of web services?**

There are four main components: XML, SOAP, UDDI, AND WSDL.

XML: eXtensible Markup Language is a uniform data representation and exchange mechanism.

SOAP: Simple Object Access Protocol, SOAP is a standard way of using XML vocabulary to enable programs on separate computers to interact across any network and describing messages between applications.

UDDI: Universal Description, Discovery, and Integration specification, UDDI is a mechanism to register and located WS based application.

WSDL: Web Services Description Language, this is a standard meta language to describe the services offered. Specifically, WSDL states what a request message much contain and what the response will look like in a clear notation. WSDL also defines where the service is available and what communications protocol is used to talk to that service.

**What is new about Web services?**

The answer is simply, XML. XML lies at the core of Web services, and provides a common language for describing Remote Procedure Calls, Web services, and Web service directories. Prior to XML, one could share data among different applications, but XML makes this so much easier to do. In the same vein, one can share services and code without Web services, but XML makes it easier to do these as well. By standardizing with XML, different applications can more easily talk to one another, and this makes software a whole lot more interesting. Try this[**course on getting started with XML**](https://www.udemy.com/learn-xml/?tc=blog.webservicesinterviewquestions&utm_source=blog&utm_medium=udemyads&utm_content=post28220&utm_campaign=content-marketing-blog&xref=blog) to master this separate language.

**What’s an example of a real web service in action?**

IBM Web Services Browser, available on the IBM Alphaworks site, is a good and intuitive example of a real web service. The browser provides a series of web services demonstrations. Behind the scenes, it ties together SOAP, WSDL, and UDDI to provide a simple plug-and-play interface for finding and integrating web services. For example, you can find a stock-quote service, a traffic-report service, and a weather service. Each service is independent, and you can stack services like building blocks.

**Explain the web services architecture.**

The operations between different software applications, which are running on a variety of platforms and frame-working are supported by web services. This web services architecture provides the concepts, modeling, and understanding web services and relationships among the components. The WSA specifies the minimal characteristics that are very common for all web services and a number of characteristics to the needed web services. WSA is called interoperability architecture that means the global elements of a global web service network are identified by this architecture in order to perform the interoperability between the web services.

**What are the steps to get a proxy object of a web service at the client side?**

There are three steps to get a proxy object of a web service at the client side.

1. Access UDDI node for a list of web services.

2. Services thus responded by UDDI have URL pointing to DISCO or WSDL document.

3. Parse DISCO and WSDL document and build a proxy object which can communicate with the web service.

**What types of operations are available in WSDL?**

There are four operations available:

1. One-way, where the operation can receive a message but will not return a response.

2. Request-response, where the operation can receive a request and will return a response.

3. Solicit-response, where the operation can send a request and will wait for a response.

4. Notification, where the operation can send a message but will not wait for a response.

**Define a REST web service.**

REST is Representational State Transfer and it is a network of web pages where the client progresses through an application by selecting links. REST is an architectural style that uses existing standards such as HTTP.