

### git\_comments:

1. Modifying a field name with an alias is backwards compatible
2. the aim to fix avro's bug <https://issues.apache.org/jira/browse/AVRO-1891> bug address explain fix the avro logical type read and write
3. Disable validation of default values for compatibility

### git\_commits:

1. **summary:** Upgrade Avro to 1.9.1 (#5938)  
**message:** Upgrade Avro to 1.9.1 (#5938) ### Motivation Currently, Pulsar uses Avro 1.8.2, a version released two years ago. The latest version of Avro is 1.9.1, which uses FasterXML's Jackson 2.x instead of Codehaus's Jackson 1.x. Jackson is prone to security issues, so we should not keep using older versions. <https://blog.godatadriven.com/apache-avro-1-9-release> ### Modifications Avro 1.9 has some major changes: - The library used to handle logical datetime values has changed from Joda-Time to JSR-310 (<https://github.com/apache/avro/pull/631>) - Namespaces no longer include "\$" when generating schemas containing inner classes using ReflectData (<https://github.com/apache/avro/pull/283>) - Validation of default values has been enabled (<https://github.com/apache/avro/pull/288>). This results in a validation error when parsing the following schema: ``json { "name": "fieldName", "type": [ "null", "string" ], "default": "defaultValue" } `` The default value of a nullable field must be null (cf. <https://issues.apache.org/jira/browse/AVRO-1803>), and the default value of the field as above is actually null. However, this PR disables the validation in order to maintain the traditional behavior.

### github\_issues:

### github\_issues\_comments:

### github\_pulls:

1. **title:** Upgrade Avro to 1.9.1  
**body:** ### Motivation Currently, Pulsar uses Avro 1.8.2, a version released two years ago. The latest version of Avro is 1.9.1, which uses FasterXML's Jackson 2.x instead of Codehaus's Jackson 1.x. Jackson is prone to security issues, so we should not keep using older versions. <https://blog.godatadriven.com/apache-avro-1-9-release> ### Modifications Avro 1.9 has some major changes: - The library used to handle logical datetime values has changed from Joda-Time to JSR-310 (<https://github.com/apache/avro/pull/631>) - Namespaces no longer include "\$" when generating schemas containing inner classes using ReflectData (<https://github.com/apache/avro/pull/283>) - Validation of default values has been enabled (<https://github.com/apache/avro/pull/288>). This results in a validation error when parsing the following schema: ``json { "name": "fieldName", "type": [ "null", "string" ], "default": "defaultValue" } `` The default value of a nullable field must be null (cf. <https://issues.apache.org/jira/browse/AVRO-1803>), and the default value of the field as above is actually null. However, this PR disables the validation in order to maintain the traditional behavior.

### github\_pulls\_comments:

1. retest this please
2. rerun cpp tests
3. Add label release-2.5.1, due to #6406 dependency
4. @massakam Thanks for this fix and looking forward for the newer Avro library to be integrated with Pulsar. Question on the fix, wouldn't the replacement of Joda-Time with JSR-310 Instant break backwards compatibility of Avro Usage within Pulsar ?
5. @codelipenghui @sijie thoughts on the above comment? ^ Wouldn't that fix break backwards compatibility with existing code base? If it does, should Pulsar also take an action of being more compliant with default Avro behaviour such as : 1. Validation of default values 2. Don't allow null values by default
6. @psilos good point. @codelipenghui @tuteng I think we need to verify the two questions that @psilos raised. If they are breaking behaviors, then it is a problem for 2.5.1. Can you guys verify that?

7. @psilos I have pushed a PR #6704 to add Joda time logical type conversion so that we can keep forwarding compatibility. And it should be included in 2.5.1. About the allow null, this is a legacy. We'd better keep forwarding compatibility. I agree with you that Pulsar schema should be compliant with default Avro behavior. We'd better draft a PIP to change the default behavior and provide options to fall back to the old behavior. I think he can plan the default value validation and allow null in 2.6.0 since 2.5.1 is on the release process.
8. @codelipenghui Thanks very much for the quick fix, and looking forward for the new release. Unfortunately one of the areas where we could have used the Avro upgrade was inside Pulsar Functions, are you planning to add the extra flag(s) into Pulsar Functions as well? Currently there is no way to override the default allow null behaviour, neither the use of JSR-310 Instant. CC @sijie
9. @psilos could you please help create an issue? so that we can keep it be tracked. Since 2.5.1 RC is out, we can plan it in 2.5.2 BTW, if you are using the Avro compiler, you need to upgrade the compiler. Otherwise, you may meet some class not found exception. Because some classes are deleted in the new Avro lib.

#### github\_pulls\_reviews:

1. I think this is not a very good way. However, there seems to be no public method to disable validation. Another option is to enable validation of default values.

#### jira\_issues:

1. **summary:** nullable fields and its default value  
**description:** as described in <https://avro.apache.org/docs/current/spec.html> {quote}h6. Unions Unions, as mentioned above, are represented using JSON arrays. For example, `{["null", "string"]}` declares a schema which may be either a null or string. (Note that when a default value is specified for a record field whose type is a union, the type of the default value must match the +first+ element of the union. Thus, for unions containing "null", the "null" is usually listed first, since the default value of such unions is typically null.){quote} and the given example for Handshake with `{"name": "clientProtocol", "type": ["null", "string"]}`, `{code}` or even the "tweet.hashtags" example from <http://stackoverflow.com/questions/31864450> as seen in <http://stackoverflow.com/questions/9417732> you have to explicitly define the `{"default": null}` for the "nullable union field", else you get `{Field clientProtocol type:UNION pos:0 not set and has no default value}` so either there is a bug in the documentation or in the code ,) If its "just" a documentation error, please decrease this bug to a wish: add some compiler warnings when using union without default value

#### jira\_issues\_comments:

1. I think the wording is somewhat confusing, but actually correct. When it says "the default value of such unions is \*typically\* null" it means that users commonly add a null default by explicitly adding it. If you don't add a default, the readers expect a value to be present. The default is only used when there is no value in the encoded bytes, but the reader expects one. The reader either gets the default that was set in the read schema, or will get an exception that there is no default value and the data file was missing the value.
2. I'm marking this "not a problem", but we can reopen if you think I'm wrong. Thanks for taking the time to report this!