

Stateful vs Stateless

~Arpit Agrawal

Hey folks!!

In my last few posts I talked about Data on K8s, why it should be used, its benefits and challenges its facing etc. But there are some Key terms which are repeated in every post & one of these key terms are **Stateful and Stateless Workloads**.

It is so rare that one has heard of K8s and not stateful and Stateless, But Why??

The answer is simple majority of users that run Data on K8s mainly uses it to run stateful workloads (mainly Databases).

If you are wondering what does these terms mean, no worries, Lets take a look

Statefulness:

Let's take an example Suppose you want to place an order on Amazon and for that you open **amazon's website; Login into your Account; Choose the product; Place an order; Do the Payment.**

Now for Every step to Happen its previous step should be remembered

Suppose if the login step is forgotten, transaction will fail or the step in which you choose your product is forgotten, again transaction will be failed.

Now it is here where the concept of **Persistence Storage** solves our problem, through persistence storage of data we can make the whole process stateful i.e., **Each step must be remembered.**

Statelessness:

This is completely opposite to Stateful, it's like a Blackboard whatever data you stored on Blackboard (in written format) is available until it has been rubbed or overwritten. This is what Statelessness is i.e., previous step is not remembered.

This is how simple these terms are yet very important.

Hope this post will help you to understand what stateful and stateless means.

This post is inspired by Kunal Verma's awesome Talk on Stateful vs Stateless on Dok day. Do check it out also <https://bit.ly/3JDmu4N>