cluedin

white paper

Why DataOps and DevOps will converge?

2019 | cluedin.com

Why DataOps and DevOps will converge?

There is no place for user interfaces in the data world. Quite a bold statement to make, but hear us out. If DevOps, Deployment, SDLC and more has taught us anything, is that a system should be stateless i.e. I should always be in a state where if I wanted to rip down a system and re-deploy it then I can do that and not lose a minute of sleep at night. This is why platforms like the Data Hub, Data Lake etc will never work in reality and production if we don't have ways to be able to version control and deploy these environments automatically.

Let's start with an example.

You are building business rules to be able to have data policies over the flow of data within your business. Due to the nice user interface of your platform you have a nice rule builder that allows you to specify simple "If this then that" types of rules. You add a few rules and save. Due to the distributed nature of systems, you now need to make sure that this change has persisted to all other environments and then you "hope" that these new rules don't put too much strain on the system. This "hope" is exactly why DevOps exists. No-one wants to work with hope anymore, we want to work with predictable, stable, repeatable and testable deployments.

Data Ops will require the same thing and it is important that we establish this straight away instead of finding out the hard way. Imagine for one moment that you had a data policy that was wrong in production. To fix this, you would want to be alerted of the problem and then you would want to immediately deploy an old version of the application and then make sure there was no need to clean any data up (or at least the application took care of the clean up for you).

Imagine you wanted to add 1000 Sharepoint sites to your production environment. You would not want to tell an admin to add them manually. You would want to prepare them in a deployment manifest and have it all automated. If we do it this way, we also will get all the versioning history, testing, deployment and publishing for free as most companies already have purchased good deployment and DevOps tools, we just need to adapt and adopt them in the DataOps environment.

Imagine then you need to remove a deprecated SAP ERP instance from your data hub. You would not want to just remove it in a user interface and then hope that everything was done perfectly, you would want to schedule a deployment which removes the integration and then runs some tests to make sure it was done correctly.

Now this can all be done through a user interface, but the reality is that in the Enteprise, it is not.

Why? Because companies are so different, and having a user interface that handles all the possible situations for all companies does not and will not exist. Rather, offering companies with the ability to augment and automate the process through well defined deployment processes allows us to control what will happen when we want it to happen. It is so much easier to accidentally do something wrong through a user interface rather than accidentally make a deployment (you would hope) go wrong.

Then again, DevOps is typically about working with the "known" i.e. you know ahead of time what needs to be deployed.

What happens when things go wrong? Will you always have the time to schedule a deployment and everything that comes along with that? Would you not want a simple toggle in a user interface to be able to quickly fix a problem. Sure. Who wouldn't want that. Hence the User Interface of Data Hub types of applications will be more catered towards this use case. Imagine a case where you forgot to place in a business rule that monitored for sensitive data. If you realised that there was a problem, you would either want to block the application and log everyone out automatically or you would want a toggle that allows you to apply the rule immediately.

I will admit. When we started CluedIn, we didn't know that this was the right approach. We got to a point where a customer was changing some settings in the UI and we thought "Oh, that could go wrong". We had definitely started out by having a clear option of "Anything you can do in the UI, you can do in the Rest API". This really helped us become compliant to this requirement, but still required some work to make it more friendly for the DevOps and DataOps team.

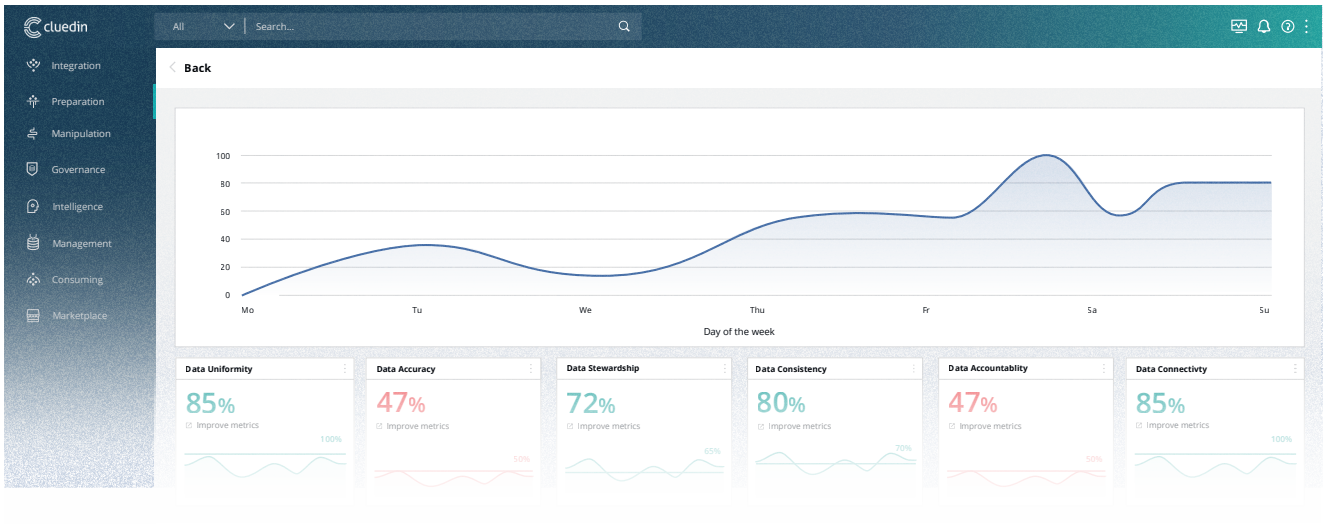
So then how do we handle the fact that we want an idempotent situation where if we need to deploy new rules that we know what it will do on the data every time.

CluedIn is not this type of system. The first reality is that ordering is not guaranteed. Here is an example, if I placed data through CluedIn today, there is no guarantee that placing it through CluedIn tomorrow will yield the same result. Why? Because our world is not idempotent and CluedIn works with live systems in production.

Imagine you are using the enrichment services from CluedIn, imagine if one of the services was operating well yesterday but was down today. This would mean that CluedIn is not able to process the data in the same way. This makes it trickier when it comes to DevOps, because there is a sense of the "unknown" when you are deploying CluedIn into your environment.

So how do we address this? We don't. Because nothing reflects the history of your data world better than real world happenings. Can you reprocess the data when that service is up? Yes. What I would rather is full logging and auditing ability that tells me that "We were not able to enrich your data from CrunchBase because it was down".

Think of it like this. If you had your code in version control and a developer accidentally checked in bad code. You would not go ahead and purge this checkin from existence, you would keep track that it happened and then fix it over the top. hat high risk data they have in file shares, emails, document repositories and more.



In summary, I know that beautiful user interfaces are nice. They demo nice and many people can easily get the concepts. In reality, it is worth recognising that they are there to solve small parts of the picture and are still relevant. It is important to remember that the main users of CluedIn will be those that are tasked with providing the rest of the business with clean, blended and governed data and hence they should know that when it comes to meeting your requirements and responsibility, CluedIn has your reality covered.

cluedin