influxdata®

AN INFLUXDATA CASE STUDY

Herrenknecht AG Uses InfluxDB to Power Its IIoT platform and Edge Data Collection

HERRENKNECHT
Tunnelling Systems

APRIL 2022

Leveraging InfluxDB At The Edge And In The Cloud Creates A Stable, Reliable IIoT Ecosystem

Company in brief

Herrenknecht is a technology leader in the area of mechanized tunneling systems. The company delivers cutting-edge tunnel boring machines for all ground conditions and in all diameters – ranging from 0.10 to 19 meters. The Herrenknecht product range includes tailor-made machines for transport tunnels (Traffic Tunneling) and supply and disposal tunnels (Utility Tunneling) as well as innovative solutions for the efficient installation of pipelines underground.

Case overview

Engineers at Herrenknecht set out to build an industrial internet of things (IIoT) platform that provided insight into live and historic data for all their tunnel boring machines (TBMs). These machines have thousands of sensors generating high velocity data, sometimes in remote areas with limited connectivity. The company chose InfluxDB to collect and manage the telemetry data for its TBMs because the platform is open source, has a range of products to accommodate growth, and runs on multiple architectures. As Herrenknecht iterated with InfluxDB, moving to InfluxDB Enterprise reduced operating costs by one-third and delivered a stable product with reliable and reputable response times for all queries.

Technologies used: Angular, InfluxDB OSS, InfluxDB Enterprise, InfluxQL, Javascript, Kubernetes, Linux, Windows

The business challenge

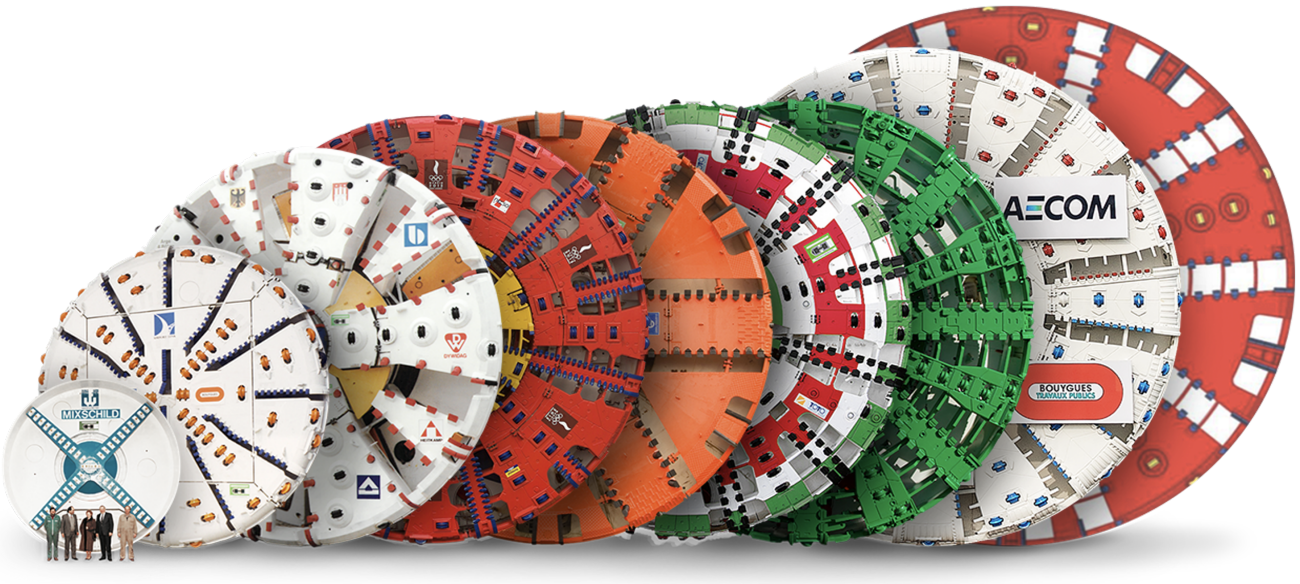
Tunnel boring machines (TBMs) are massive industrial machines. The cutting surfaces for these machines have a diameter that ranges from 0.10 to almost 20 meters. As the size of the machines grew, so too did the amount of data Herrenknecht needed to capture about their operation; this is especially true for machines used in traffic tunneling. Having current and accurate information is critical for a TBM operator so they know what to do, how to direct the machine, and how to react to sudden events.

The Herrenknecht team wanted to build a platform that provided this type of data to TBM operators, site foremen, project managers, and executives. In order to reduce costs, they wanted a solution based on open source software. They also needed to build a solution that their small team could easily manage and maintain. The team wanted to focus their time on developing features and not on operations and DevOps tasks.

The technical challenge

The Herrenknecht team had a number of challenges to address when building its IIoT platform. First, the platform needed to support more than 2,000 TBMs. Of these, several hundred could be working simultaneously in the field at various places around the globe. A single TBM can have 5,000 sensors, and this number increases as the company builds newer machines. Furthermore, each sensor can have a different sample rate. Some sensors have a rate as low as 100 milliseconds. Herrenknecht collects data to feed cutting surface metrics like advance speed, advance progress, cutting wheel torque, maximum allowed penetration per minute, and maximum allowed thrust force.

Connectivity can be another challenge. Operators deploy TBMs at job sites all over the world, often in remote locations, so the Herrenknecht team needed to factor in connectivity and bandwidth issues with machines in the field. The very nature of tunneling means that these machines can also work 10–15 kilometers deep, which can create connectivity issues sometimes for days, weeks, or even months where a TBM is completely offline and disconnected from the internet.



1985	1996	1997	2006	2006	2010	2016	2013	Concept
HERA	**Sydney**	**Hamburg**	**Madrid**	**Shanghai**	**Sparvo**	**Santa Lucia**	**Hongkong**	**St. Petersburg**
5.95 m	10.70 m	14.20 m	15.20 m	15.43 m	15.62 m	15.87 m	17.6 m	19.25 m

Source: Herrenknecht

Furthermore, the Herrenknecht team wanted to be able to include insights from historical data in the new platform. The company collected data from machines for over four decades and wanted to make that data available. This data existed in many different formats, including in SQL databases, and DBX and CSV files. Therefore, they needed a solution that would be able to ingest and handle data from these varied sources.

Finally, Herrenknecht wanted to future-proof their solution as much as possible. They required a platform that would work for 10–15 years, as they did not want to reinvent the wheel every few years. Therefore, they needed a stable system, but one that was also able to grow with the company's needs. This required a robust and flexible architecture that allowed them to exchange components at any time without a lot of effort or outages.

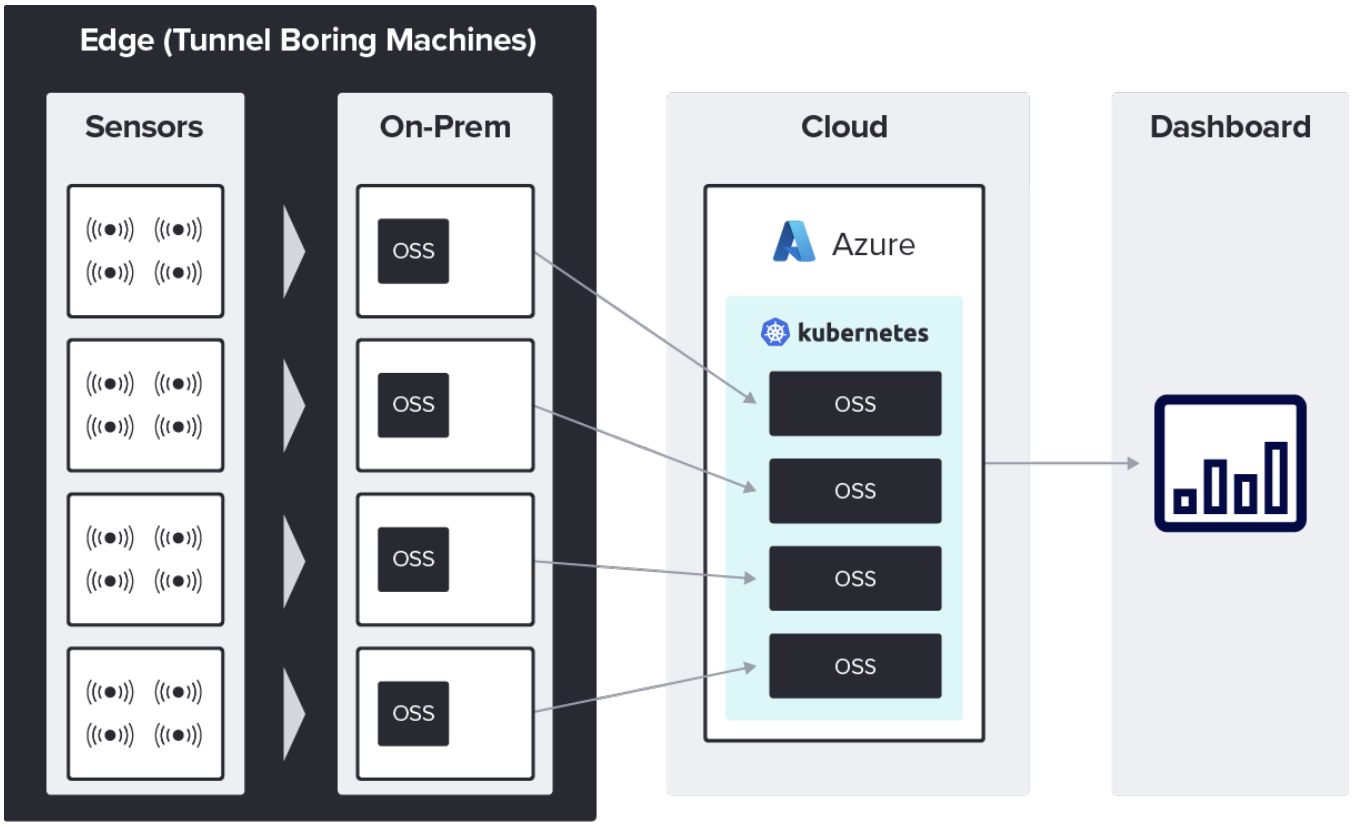
The solution

Herrenknecht developers chose InfluxDB as the central time series storage database for their IIoT platform. Several factors played into this decision. The most important was that InfluxDB offered both an open source version and a commercial enterprise edition that accommodated growth. Herrenknecht's technology stack played a role in the decision as well. At the time, the TBMs ran on Windows so the fact that InfluxDB has a Windows build was critical. Now, the TBMs run on a mixed environment of Linux and Windows, but the instances of InfluxDB continue to run on Windows.

Herrenknecht developers take a somewhat unique approach to data storage aimed at reducing memory usage and maintaining low series cardinality for its time series data. Instead of collecting measurements for individual sensors or for specific sensor groups, they collect one measurement that has thousands of fields, where every field is a float type. They round data values to determine if a significant change occurs and only write values that qualify as a significant change. Sometimes the team needs to consider the accuracy of the sensor in question and re-round if the detected change is greater than the accuracy threshold. If a significant change does not occur, they simply remove those duplicate values. They use tags sparingly, primarily to indicate the operational state of a TBM and these tag value types are simple Booleans. Using this approach, Herrenknecht ends up writing, on average, between 1–5GB of data per machine, per month.

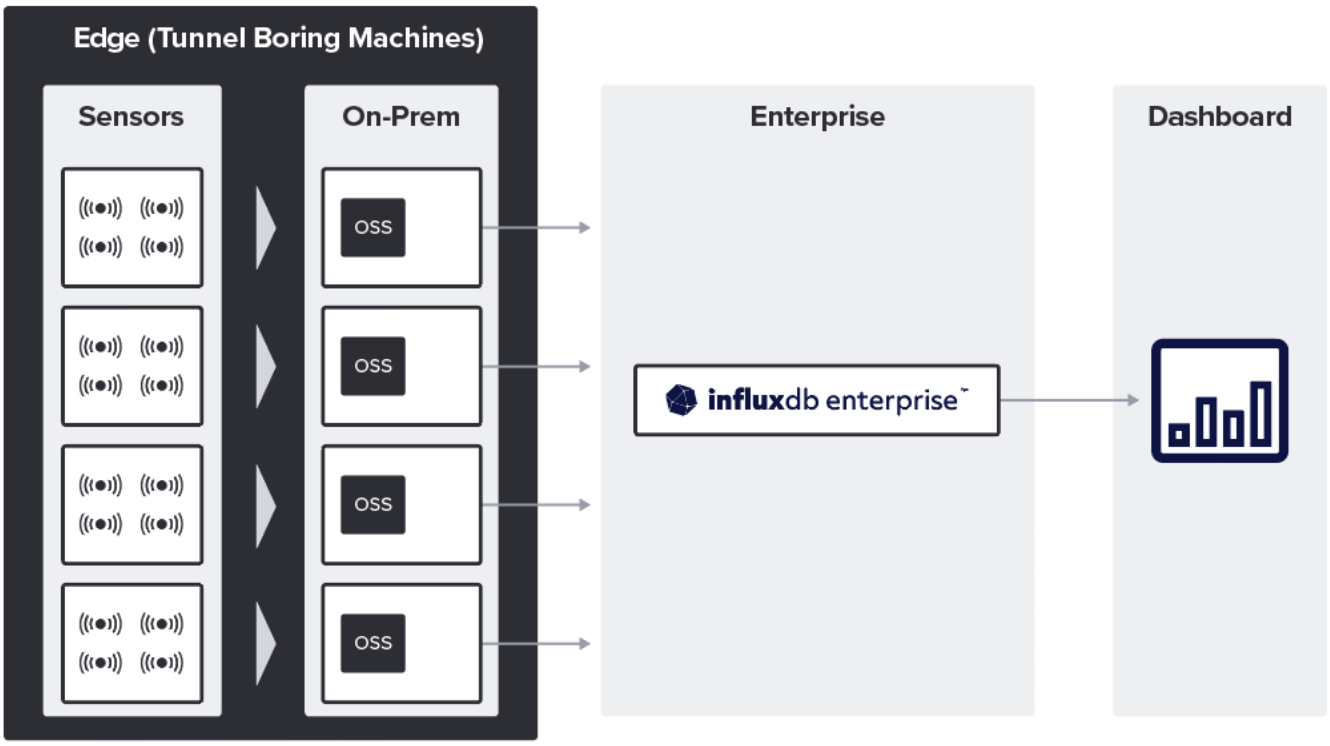
Edge devices

Herrenknecht gets data into InfluxDB using a C#-based client library and then stores data on-site, directly on the TBMs, so each machine runs an instance of InfluxDB. This allows Herrenknecht developers to take advantage of InfluxDB's data processing and storage capabilities. They use InfluxQL for data processing and custom software to derive aggregations from their granular data. Their services run on top of those aggregations. For customers that do not want to copy their data to the cloud, the edge instances of InfluxDB power dashboards locally.



InfluxDB OSS in the cloud

Next, Herrenknecht developers built a cloud solution using InfluxDB OSS. For this, the Herrenknecht team opted to run InfluxDB inside Kubernetes, with one InfluxDB container per TBM and Microsoft Azure Cloud for storage. This approach is almost identical to what they run on the TBMs so there was very little additional development necessary. When it comes to transferring on-site data to the cloud, this setup has additional benefits. Because they run a local instance of InfluxDB on the machine itself, which does all the data cleaning and processing, they only need to send the clean data to the cloud. The Herrenknecht developers built a custom synchronization based on a custom REST API so the data in the cloud mirrors exactly the data on the TBM. This is critical because the visualizations in the cloud and on the machine need to be identical.



Enterprise

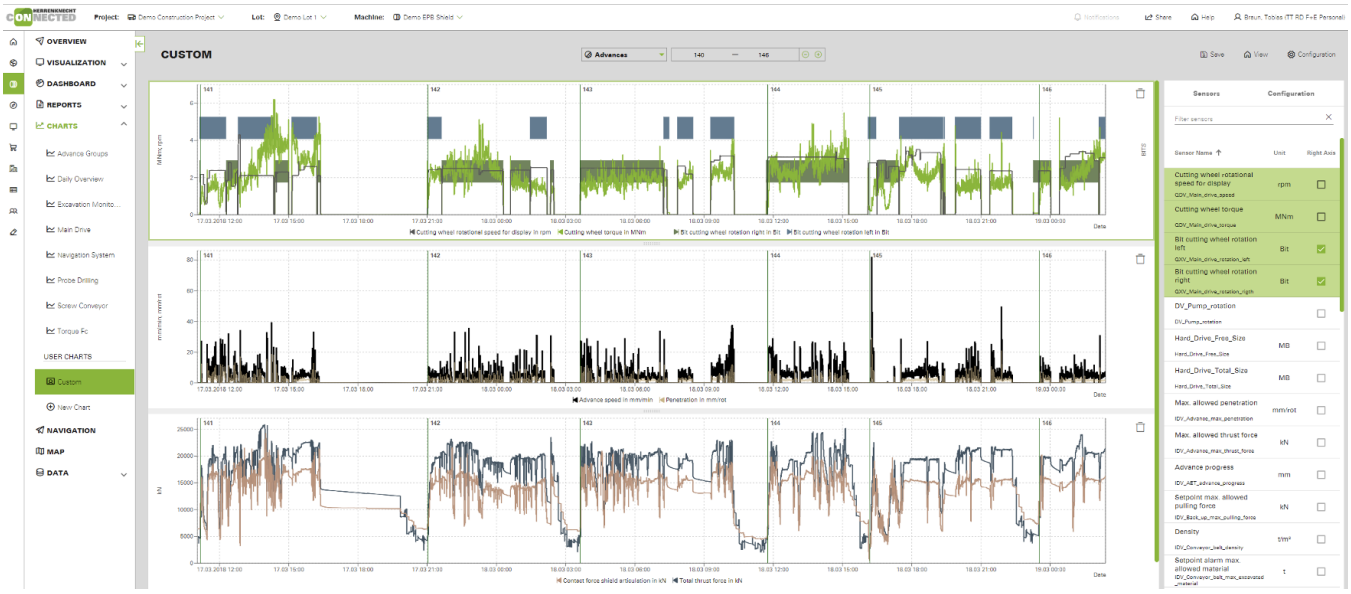
The cloud system described above worked well until Herrenknecht reached approximately 100 instances of InfluxDB. Issues with Azure storage created reliability issues, delayed queries, and increased response times. The company decided to migrate from running open source in the cloud to InfluxDB Enterprise to improve the system reliability. Because they already had a large store of historical data, Herrenknecht developers sought to migrate that data to the Enterprise version with minimal downtime, at most a few minutes per machine. The team completed the migration in less than two weeks, moving from the Kubernetes clusters to the InfluxDB Enterprise clusters with no read downtime and less than sixty minutes of write downtime.

> *We have the open-source version of InfluxDB on every tunnel boring machine that we deliver, and we have InfluxDB Enterprise in the cloud, and it just works in the background. We don't really have to take a lot of care with it. It's just there, and it's reliable, so we don't have a lot of DevOps efforts ... and our small team can concentrate on feature development.*

Tobias Braun, Software Architect, Herrenknecht AG

Results

Following the migration process, the total cost of ownership of Herrenknecht's InfluxDB system went down by one-third. The main factor for this savings came from running fewer virtual machines. Instead of having many Kubernetes nodes for all the InfluxDB containers, they switched to a smaller, more cost-effective cluster of InfluxDB Enterprise.



Source: Herrenknecht

This transition also delivered the reliability and stability that Herrenknecht needed to power its IIoT platform. The switch eliminated the issues the company encountered with slow queries and increased response times. The team built a custom front-end Javascript UI based on Angular for their platform that pulls data directly from InfluxDB. They also wrote a custom API to enhance and augment the data in InfluxDB to feed additional visualization options for TBM operators. Perhaps most importantly, the company's customers are happy with the IIoT platform and interface.

To learn more about Herrenknecht AG, visit their website.

About InfluxData

InfluxData is the creator of InfluxDB, the leading time series platform. We empower developers and organizations, such as Cisco, IBM, Lego, Siemens, and Tesla, to build transformative IoT, analytics and monitoring applications. Our technology is purpose-built to handle the massive volumes of time-stamped data produced by sensors, applications and computer infrastructure. Easy to start and scale, InfluxDB gives developers time to focus on the features and functionalities that give their apps a competitive edge. InfluxData is headquartered in San Francisco, with a workforce distributed throughout the U.S. and across Europe. For more information, visit influxdata.com and follow us @InfluxDB.