Lucid

Lucidscale

Observian

Observian

Observian uses Lucidscale to coordinate across teams in order to design new and optimized AWS, GCP, and Azure cloud solutions for customers.

Key benefits

GAIN VISIBILITY INTO EXISTING ARCHITECTURE

Understand current states and gain alignment with customers.

DESIGN NEW AND OPTIMIZED SOLUTIONS

Work towards new—or optimized—solutions for customers' AWS, GCP, or Azure architecture.

PREVENT TRIBAL KNOWLEDGE

Mitigate the risk of knowledge unintentionally belonging to a single person.

IMPROVE ONBOARDING FOR ARCHITECTS AND ENGINEERS

Centralize essential information to get new engineers and architects up to speed faster.

INDUSTRY: Tech
SIZE: Small (1-100 employees)
ROLE: Professional Services, IT

Observian is a cloud consulting firm that offers cloud services such as cloud migration, disaster recovery, security, compliance, cost optimization, and more.

OBSERVIAN

In an era that deems digital transformation a top priority for businesses, it can be challenging to find the time and resources to overhaul and optimize outdated systems and cloud processes.

Observian is a cloud consulting firm that helps companies improve their cloud operations. They provide cloud services and software delivery with professional and managed service offerings. Think everything from re-architecture support to application modernization projects ranging from networking and applications to cloud and database migrations.

In order to keep up with the influx of clients, Observian needed a solution that would help them work faster and smarter, preferably a visual solution. They use Lucidscale to gain visibility into customers' existing architecture, quickly deliver professional architecture diagrams that help customers understand their own infrastructure, and better onboard and manage knowledge transfer internally.

Gain visibility into existing architecture

Before Observian can make recommendations and execute plans, they need to gain insight into what their customers' architecture currently looks like.

But digging deeper into customers' architecture requires a forensics mindset to discover what's been built over the years—sometimes even decades. "We have to research and comb through many different pieces in their environment. And it's not just their infrastructure or application—it's their network, it's everything that's connected," said Scott Plamondon, co-founder and VP of architecture at Observian.

When you're dealing with paying clients, however, time is of the essence. "You can imagine the customer ends up paying more on the professional services side because we need to spend more time with them getting as much insight as we can," said Scott.

Cloud architecture diagrams are essential to understanding current state and planning future state, but according to Aravind Marthineni, solutions architect associate at Observian, there's a very good chance customers don't have an updated cloud diagram—as often as 9 times out of 10.

If clients do have diagrams, it's usually an afterthought. "They'll do it for an audit. Or when they're designing a new product," said Scott. But he adds, "Once the product or feature has been started on, they'll say, 'Hey, we were planning to do it that way, but we totally did another thing, but this is the diagram we're going to send you."

Client diagrams are usually an outdated version posing as current, failing to account for—or document—justified changes that were made throughout the development cycle—even if those changes were made within the last 24 hours. Because the cloud moves faster than traditional on-prem environments, cloud diagrams can become out of date quickly.

As part of the consultation and partnership with clients, Observian creates these cloud diagrams to plan and verify infrastructure changes for Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure. But building them is notoriously time-consuming and lacking in proper detail.

Tackling even one or two cloud architecture diagrams manually is a lot of work. And Observian deals with way more than that. "We might land three new customers in a single day and all of them want cloud diagrams and different aspects of the same diagram. Lucidscale has been a need of the hour. There was nothing that could directly import architecture from the cloud, and Lucidscale is definitely a pioneer in that field," said Aravind.

> "I used to spend anywhere between four to seven hours on a diagram for a customer, whereas now I could just pull it and make some tweaks, maybe change certain views, and send it over in less than 15 minutes."

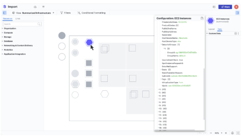
Not only does Lucidscale help Observian quickly auto-generate diagrams of their customers' AWS, GCP, or Azure cloud architecture, but it also allows them to access metadata in the context of a diagram, rather than toggling between cloud console and diagram. Because Lucidscale diagrams are backed by metadata brought in from AWS, GCP, and Azure, Observian can visualize a customer's cloud from multiple views, such as through network rules or as an overview of accounts, regions, and VPCs. This makes it easier for the Observian team to understand exactly what their customers' environments look like and more easily deliver cloud diagrams and different aspects of the same diagram that clients want, like Aravind mentioned.

Observian uses the Lucidscale filters to create tailored views of the diagram to dive into specific pieces of the infrastructure to address details. As an example, Scott said, "When customers say they've changed, added, or removed services, I can actually see what they did and get a little better insight. Is it the same as actually being in the environment? No, but it's enough that gets me moving in the right direction or gets our team moving in the right direction."

Lucidscale is integrated directly into Observian's engagements to conduct well-architected reviews, which Aravind uses to review findings and recommendations with customers.

> "Cloud architecture is an ever-evolving landscape—it's always changing. Lucidscale really helps us keep track of everything and ship it to customers in a deliverable that is very neat," said Aravind.

After setting the stage with customers, the Observian team can confidently move forward with building what customers came for: new and optimized solutions.



Design new and optimized solutions

Observian uses both Lucidchart and Lucidscale together constantly throughout the process as they work towards new—or optimized—solutions for customers' AWS, GCP, or Azure architecture.

Their workflow looks a little something like this:

- Understand and verify current environments (Lucidscale).
- Design a new/optimized solution (Lucidscale, Lucidchart).
- Use a future state diagram to get approval from the customer (Lucidchart).
- Use a future state diagram as an implementation guide (Lucidchart).
- Refresh the diagram to verify implementation was done correctly (Lucidscale).

According to Scott, Observian uses Lucidchart to diagram what they're going to build for them, so everyone is aligned on the future state plans. Those same Lucidchart diagrams guide building and implementation, ultimately quickening delivery time to clients.

Rather than stamping and sending it, hoping what was built meets expectations, Observian uses Lucidscale to validate the build was done correctly and according to plan. If not, they can drill into the information that matters most and take action where needed, repeating the process over until the desired results are achieved.

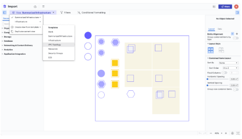
Prevent tribal knowledge

Not only does Observian use Lucidscale to save time, create a shared understanding, and plan and verify changes, but they also use it to prevent the risk of knowledge unintentionally belonging to a single person—often described as "tribal knowledge." The risk is that if an employee leaves the company, the knowledge goes with them, which can significantly delay or hinder internal processes and customer relationships.

Even on a five-person operational team, it's likely that only one or two people truly understand the full infrastructure, while the responsibility of other team members—probably junior developers—is to fix bugs and make small infrastructure changes.

But if tragedy strikes, someone is out sick, or someone goes on vacation, and an infrastructure incident occurs, then what?

> "If a huge infrastructure failure happens, those junior devs have no idea which component of the infrastructure failed. You will need something to look at. And that's one big reason why diagrams are important. And not only in crisis situations—but also how teams transition over a period of time," said Aravind.



Improve onboarding for architects and engineers

To ensure no single employee has all the knowledge and answers, essential information needs to be easily accessible to new engineers and architects. "Each time a new member comes in, some person has to transfer knowledge to new employees, resulting in time lost in man-hours," said Aravind. "[With Lucidscale,] you can document your entire infrastructure in a diagram and pin it to documentation and keep it updated."

Rather than relying on senior employees to onboard new engineers and architects, they can rely on infrastructure diagrams that stay updated on their architecture team Confluence page using the Confluence integration. "And then, when they have doubts, they can come to the team lead or the manager of that team to clarify things from there," said Aravind.

Scott attributes a great deal of their success to white-glove customer management and customer relationships built on speed, transparency, and trust—no matter the project. Before a project starts, they use architecture diagrams to ensure internal alignment and that customers fully understand the timeline, the offering, and the exact path to achieve success.

With the help of Lucidscale, Observian makes strategic recommendations and delivers quality results that customers understand and can maintain. Because when teams and customers are on the same page, you can turn cloud insights into cloud success: the Observian promise.