**Amazon Leadership Principle: Customer Obsession**

**Story Title**: *New Logo Acquisition Dashboard at Equinix*

**S/T – Situation and Task (~1 minute)**

When I joined Equinix, one of the major gaps I identified was that the sales team lacked visibility into new customer acquisitions—what we call "new logos." There was no centralized way to track or report on these wins, which made it difficult for leadership to assess sales performance or strategize growth. I realized that without clear tracking, we were losing insights into one of the most critical indicators of business expansion. I took the initiative to build a comprehensive solution that would allow us to measure new logo performance across time, regions, and products.

**A – Actions (~3–5 minutes)**

I started by setting up stakeholder meetings with regional sales heads, product owners, and the Salesforce operations team to understand how they currently define and perceive "new logos." The definitions varied widely, which created misalignment. I facilitated discussions to standardize the KPI: we agreed that any opportunity marked "won" under a previously unengaged parent account hierarchy would be defined as a "new logo."

Once we finalized the definition, I began building the data pipeline. I extracted opportunity and account data from Salesforce using SOQL queries and loaded it into our data warehouse using an ETL process. I created data models in SQL to aggregate new logo data by quarter, region, sales segment, and product. Then I collaborated with our Tableau team to design a highly interactive dashboard.

The dashboard featured KPIs such as:

* **New Logos YoY, MoM, and QoQ**
* **Average Deal Size per New Logo**
* **New Logos by Product Line and Sales Segment**
* **Top Performing Regions (IBX-based)**
* **Salesperson-level performance**

I also conducted enablement sessions with the sales teams to ensure they understood how to use the dashboard to drive their strategy and uncover gaps.

**R – Results (~1–2 minutes)**

Within four quarters of rollout, we observed a **12% increase in new logo acquisition** YoY—one of the highest upticks in recent years. Sales teams reported the dashboard helped them track performance gaps and realign their outreach to untapped accounts. Leadership started referencing the dashboard in quarterly reviews, and it became a key input in regional goal-setting.

This project strengthened cross-team collaboration and earned me recognition from sales leadership for "improving strategic visibility into core growth metrics." More importantly, it shifted our mindset from reactive to proactive when it came to customer acquisition tracking.

**Amazon Leadership Principle: Customer Obsession**

**Story Title**: *Inbox Dashboard Automation Using Python & Smartsheet*

**S/T – Situation and Task (~1 minute)**

At Equinix, I noticed a recurring issue: although we had over 15 business-critical dashboards created for different teams, stakeholders—due to their busy schedules—were not regularly logging into Tableau to check them. This lack of engagement meant critical insights were being missed or delayed, and that weakened the very purpose of those dashboards. I recognized the need for a solution that would make it easier for users to access the right insights *without changing their routine*—essentially, bring the data to them.

**A – Actions (~3–5 minutes)**

I decided to create an **automated inbox delivery system** that would let stakeholders receive tailored dashboard snapshots directly in their email—based on their preferences.

I started by designing a Smartsheet form where stakeholders could:

* Select which dashboards they wanted to receive
* Apply filters (e.g., region, time period, sales segment)
* Choose delivery frequency: daily, weekly, or monthly

This form automatically fed into a centralized Smartsheet table where all preferences were logged.

On the backend, I built a **Python script** that integrated with Tableau Server. The script:

* Queried Smartsheet for active user preferences
* Triggered Tableau’s REST API to render customized dashboard views
* Captured and saved snapshots of filtered dashboards
* Emailed the snapshot along with a dynamic link to the live dashboard

I also added logic to include summary stats in the email body to give stakeholders a quick glance without opening the full dashboard.

Before launch, I conducted A/B testing by selecting two pilot teams: one continued using the dashboards manually, while the other used the inbox automation. I tracked engagement through Tableau Server usage metrics and click rates.

**R – Results (~1–2 minutes)**

After full rollout, the dashboard automation system led to a **70% increase in stakeholder engagement** with Tableau dashboards. Email open rates exceeded 80% weekly, and over 60% of users clicked through to the live dashboards at least once per week. Stakeholders reported that having dashboards delivered to their inbox improved their ability to make timely decisions.

This automation became a standard delivery mechanism for all executive-level reporting and even inspired similar mechanisms for marketing and finance teams. It also reduced the need for BI follow-ups and dashboard reminders from analysts.

**S/T – Situation and Task (~1 minute)**

During the December holiday period at Equinix, most of my BI teammates were on extended leave. This coincided with a time when the business typically needed year-end performance reports, dashboard updates, and backlog cleanup—especially from sales, finance, and operations. There was no formal handover plan in place, and without intervention, these requests would have been delayed, impacting decision-making and team trust.

Recognizing the potential disruption, I proactively volunteered to take full ownership of BI responsibilities during the break to ensure continuity and stakeholder satisfaction.

**A – Actions (~3–5 minutes)**

First, I assessed the open items in our **Jira board**, focusing on active epics and high-priority tickets tagged for month-end and quarter-end reporting. I also reviewed upcoming tasks that were assigned to others but couldn’t wait until the team returned.

I took the following ownership-driven actions:

* **Managed Jira pipeline**: Triaged and re-prioritized 15+ incoming tickets and enhancements, reassigning them to myself based on urgency and business impact.
* **Stakeholder coordination**: I reached out to stakeholders directly (including Sales Ops, Finance, and Marketing) to clarify outstanding requirements and timelines, ensuring no assumptions or misalignment.
* **Delivered enhancements**: Fulfilled several Tableau dashboard updates and data quality fixes for executive reports that were scheduled for QBRs.
* **Handled ad hoc reporting**: Built new visualizations and ad hoc SQL queries for a last-minute sales incentive analysis request from regional leadership.
* **Maintained documentation**: Updated internal documentation so my team could seamlessly pick up where I left off after the holidays.

Throughout this period, I acted as the single point of contact for BI across multiple business units and ensured nothing fell through the cracks—even without direct supervision or escalation.

**R – Results (~1–2 minutes)**

As a result of this ownership, **100% of December BI tasks were delivered on time**. I received direct appreciation from multiple stakeholders for my responsiveness, attention to detail, and business understanding.

Our director later cited this during our team-wide retrospective as a best-practice example of stepping up during a high-risk period. It strengthened stakeholder trust in our BI team and also led to a formal **holiday support coverage process** being adopted based on my approach.

**Amazon Leadership Principle: Ownership**

**Story Title**: *IA Request Tracker Using Smartsheet*

**S/T – Situation and Task (~1 minute)**

At Equinix, our team received a constant stream of IA (Insights & Analytics) requests—including dashboard enhancements, data model changes, and ad hoc reporting—from various business units like Sales, Marketing, and Operations. However, these requests were submitted inconsistently: some came through Teams messages, some via email, others were mentioned casually on calls. We were using a combination of Excel and verbal updates to track them, which led to **frequent misses**, poor visibility, and stakeholders having to follow up multiple times to ask for status updates.

Recognizing that this fragmented intake process was creating frustration and eroding trust, I took ownership of solving it—even though no one had formally asked me to.

**A – Actions (~3–5 minutes)**

I started by auditing our previous IA requests over the last 3 months and found that nearly **30% had no formal record** or tracking. I also spoke with stakeholders who shared frustration about status confusion or having to repeat requests.

To fix this, I designed and implemented a **centralized IA Request Tracker using Smartsheet**. Here’s what I did:

1. **Built a structured Smartsheet Form** that allowed stakeholders to submit requests with:
   * Type of request (New dashboard, enhancement, ad hoc, data model field addition)
   * Priority level (High/Medium/Low)
   * Desired delivery date
   * Contact info and business justification
2. **Automated workflows**: I connected the form to a live Smartsheet tracker with automated alerts to notify the team when a new request came in. I also set up logic to assign tasks based on request category.
3. **Created visibility**: Added status columns like “In Progress,” “Under Review,” and “Completed” so both the BI team and stakeholders could see real-time updates.
4. **Trained the team and business users**: I ran walkthrough sessions with stakeholders to introduce the form and shared usage instructions. I also added the tracker link to our email signatures and Teams channel so it became the default intake method.
5. **Monitored adoption**: Over the next 8 weeks, I tracked submission trends and usage analytics to make small iterative improvements, like adding optional attachments and auto-acknowledgment messages.

**R – Results (~1–2 minutes)**

After implementing the IA Tracker, **missed or lost requests dropped by over 80%**. Every new request was tracked and addressed within an expected SLA window. Stakeholders appreciated the transparency and no longer had to follow up repeatedly.

The Smartsheet tracker became a **standard operating process** for all IA submissions across our team. This also saved analysts time otherwise spent manually organizing requests and reduced confusion over prioritization. Most importantly, it helped **restore stakeholder confidence** in our responsiveness and delivery reliability.

**Amazon Leadership Principle: Invent and Simplify**

**Story Title**: *Replacing Manual Excel-Based Revenue Reports with a Live Tableau Dashboard*

**S/T – Situation and Task (~1 minute)**

At Equinix, our team was responsible for reporting key revenue metrics like **Revenue by Product**, **Revenue by Country**, and **YoY Revenue Growth** to senior leadership. However, the existing process involved **downloading raw data from multiple systems**, manually merging Excel files, applying complex formulas, and creating charts every quarter.

This approach was not only time-consuming, but also **error-prone, inconsistent**, and hard to replicate when leadership asked for past quarter comparisons. It consumed hours of analyst time and lacked version control, which created confusion.

Recognizing that this manual process was unsustainable and inefficient, I took the initiative to **invent a simplified, automated, and scalable dashboard solution** that could deliver real-time revenue insights with zero manual prep.

**A – Actions (~3–5 minutes)**

To eliminate the need for manual Excel merging, I first audited the entire data pipeline and reporting logic:

1. **Identified Source Tables**: I located and validated data from our booking system, product catalog, IBX (data center location), and country reference tables.
2. **Data Modeling & Integration**: I joined these tables in the data warehouse using SQL to ensure consistent keys across dimensions like product, region, and time. I also applied filters and calculations to define key KPIs like:
   * Revenue by Product and Country
   * YoY Growth
   * Regional Revenue Trends (IBX)
   * Quarterly and Monthly Breakdown
3. **Automation**: I created a live Tableau data source connected directly to the warehouse and scheduled **daily automatic refreshes**. This removed the need for manual data pulls or rework every time a report was needed.
4. **Dashboard Design**: I designed a highly interactive Tableau dashboard that allowed users to:
   * Select filters (Time Period, Product, Country, IBX)
   * View charts and KPI summaries
   * Export visuals for leadership presentations
5. **Change Management**: I conducted enablement sessions for analysts and stakeholders to transition them away from Excel and toward the new dashboard. I also embedded the dashboard in our BI portal, making it easily accessible company-wide.

**R – Results (~1–2 minutes)**

The new Tableau dashboard completely **eliminated the need for Excel-based reporting**, reducing manual effort by over **90%**. Analysts saved several hours per week, and errors in formula-based calculations were eliminated.

Leadership had **live access to revenue metrics** at any time, with improved accuracy and visual clarity. The dashboard was adopted by Sales Ops, Finance, and Strategy teams and became the **single source of truth** for revenue reporting.

This project significantly **improved data trust and efficiency**, and it set a foundation for expanding self-service analytics to other teams.

**Amazon Leadership Principle: Invent and Simplify**

**Story Title**: *Automated New Logos Validation Using Excel VBA for Salesforce QA*

**S/T – Situation and Task (~1 minute)**

At Equinix, our BI team was responsible for validating daily “new logo” entries coming from Salesforce. The business logic was strict: only **first-time opportunity wins**, with **positive MRR values** (direct > 0, indirect ≥ 0), **contract terms over 12 months**, and **booking dates within the current month** would qualify.

Each day, the team downloaded Salesforce data, applied 5–6 manual filters in Excel, and then compared the results with the official Tableau dashboard. If there were mismatches, we would submit those to the IT team for correction in Salesforce. This **manual QA process took 30+ minutes daily**, was repetitive, and error-prone due to formula misapplication.

I saw an opportunity to **simplify this process with automation** so analysts could focus on higher-value work instead of mechanical filtering.

**A – Actions (~3–5 minutes)**

I designed and built a **VBA macro automation** to replicate the full validation logic inside Excel using a simple “one-click” solution. Here’s how I approached it:

1. **Mapped the business rules**: I codified the logic—first win opportunity, MRR filters (direct > 0, indirect ≥ 0), contract term > 12, and booking month = current month—into VBA if-statements and filter logic.
2. **Macro recording and logic scripting**: I started with Excel's macro recorder and then manually enhanced the script to dynamically apply filters, perform row-by-row checks, and flag invalid entries.
3. **Generated summary output**: The macro saved a clean, formatted file with the **“invalid new logos”** that needed correction, so the team could directly send it to IT.
4. **Built for usability**: I added a **button in the Excel ribbon** labeled “Run New Logo Validation” so anyone on the team could execute it without writing a single line of code.
5. **Error handling and edge cases**: I implemented checks for missing columns and empty rows, and built logic to adjust to variable file lengths and formats.

I also documented the tool and shared a quick training video so others could use and maintain it.

**R – Results (~1–2 minutes)**

The VBA tool **reduced the manual validation task from 30+ minutes to under 2 minutes** per day and eliminated human errors caused by inconsistent filtering or formula mistakes.

It saved the team more than **120 hours per year**, and improved the accuracy of new logo records used in executive-level reporting. The output file also standardized IT submissions, reducing back-and-forth on what needed to be corrected in Salesforce.

The tool became a **daily-use asset** for our BI QA workflow and served as a model for automating other validation checks later.

**Amazon Leadership Principle: Are Right, A Lot**

**Story Title**: *Booking Logic Redesign — Long-Term vs Short-Term Contracts*

**S/T – Situation and Task (~1 minute)**

At Equinix, our sales and finance stakeholders relied on Tableau dashboards to track booking metrics — specifically, how much revenue was being secured in our sales pipeline. However, I noticed that the current dashboards **combined both short-term and long-term contracts**, treating them equally in the reported booking values.

This was misleading. A short-term contract (e.g., 3–6 months) doesn’t represent the same financial value or customer commitment as a long-term one (12+ months). Including both in one metric **gave stakeholders a distorted view** of our pipeline’s health and long-term growth.

Even though the dashboards were already in active use, I believed we needed to **correct the logic and enable stakeholders to differentiate** between short- and long-term bookings clearly.

**A – Actions (~3–5 minutes)**

I started by reviewing our existing data model and confirmed that the dashboard logic used a general sum of all contract MRR, regardless of contract term length. I pulled examples of short-term bookings inflating quarterly numbers and presented those to leadership to explain why this was a risk.

Then, I proposed a solution:

* **Split the logic** to separate long-term bookings (contract term > 12 months) from short-term bookings (contract term ≤ 12 months)
* **Create a user-facing parameter** in Tableau so stakeholders could toggle views:
  + Long-term only
  + Short-term only
  + Combined view

This would make it transparent, flexible, and remove ambiguity.

On the backend, I implemented logic in SQL to flag bookings based on contract term thresholds and updated the Tableau dashboard to include:

* Two new KPIs: *Long-Term Booking Amount* and *Short-Term Booking Amount*
* A visual indicator (color-coded bar) to highlight when short-term bookings exceeded thresholds
* A tooltip explaining the difference so users understood how metrics were derived

To ensure adoption, I conducted live walkthroughs with regional sales leads and finance partners and answered questions around how this change would impact reporting and historical comparisons.

**R – Results (~1–2 minutes)**

The new logic and toggle view were well-received and immediately adopted by leadership. Stakeholders appreciated the clarity, especially during quarterly planning and board-level reporting where long-term revenue stability is a key metric.

The solution **prevented misinterpretation of data**, and helped sales teams prioritize deals with higher strategic value. This also **influenced forecasting** by enabling leadership to separate “quick wins” from long-term revenue builders.

The dashboard became a **trusted source for strategic planning**, and my change was cited in a quarterly BI retrospective as a great example of applying business judgment to improve data quality.

**Amazon Leadership Principle: Are Right, A Lot**

**Story Title**: *Clarifying Complex KPIs Using Tooltips in Book-to-Bill Dashboard*

**S/T – Situation and Task (~1 minute)**

At Equinix, I was part of the BI team building a **Book-to-Bill dashboard**, which aimed to visualize how long it takes from customer booking to billing. The dashboard tracked metrics like average time to bill by product, delays across IBX metros, and product-level cycle time.

However, after launch, I noticed a recurring issue: **stakeholders struggled to understand the KPI definitions**. Even though the numbers were accurate, they often misinterpreted what the visuals represented — leading to repeated clarification requests and, in some cases, incorrect assumptions during meetings. I realized that if people didn’t trust or fully understand the metrics, the dashboard would fail its purpose.

So I took it upon myself to **solve this misalignment between data and comprehension**, using a simple but thoughtful approach.

**A – Actions (~3–5 minutes)**

Instead of redesigning the dashboard or simplifying the KPIs (which were business-critical and needed to stay intact), I focused on improving **context and communication** through user experience design.

Here’s what I did:

1. **Audited the questions** we received post-launch — most of them were around KPI definitions: “What does this metric really mean?”, “Does this include indirect billing?”, “Why is this product showing high delay?”
2. Proposed adding **interactive tooltip icons** on every visualization — allowing users to hover and instantly view:
   * Plain-language KPI definitions
   * Formula explanations (e.g., how booking-to-bill time is calculated)
   * Notes on exceptions or exclusions (like indirect vs. direct)
3. I created tooltip templates and worked with stakeholders to validate the definitions were clear and accurate.
4. Implemented the feature in Tableau using custom shapes and hover-enabled text objects.
5. Pushed this change to production, then monitored user feedback and dashboard engagement metrics for two weeks.
6. Based on the positive response, I **standardized this tooltip feature** across all high-impact dashboards in our BI ecosystem.

**R – Results (~1–2 minutes)**

After implementing tooltips, we saw a **sharp drop in stakeholder confusion and follow-up questions** — in fact, clarification requests on Book-to-Bill metrics dropped by **over 60%** within the first month.

The dashboard gained wider adoption because users could now understand what they were seeing without needing to consult the BI team. This **freed up analyst time** and increased **stakeholder trust** in the reporting.

More importantly, this validated my instinct: I recognized that *being technically right* wasn’t enough — **being clear and trusted was just as important**. The tooltip model became a best practice and was later adopted in other dashboards across Sales Ops and Finance.

**Amazon Leadership Principle: Insist on the Highest Standards**

**Story Title**: *Dashboard Performance Optimization – From 80s to 10s Load Time*

**S/T – Situation and Task (~1 minute)**

At Equinix, several critical dashboards—including Sales Revenue, Funnel, and Fabric Attach Rate—were **performing poorly**, taking up to **80 seconds to load**. These dashboards were used regularly by Sales Ops and executive leadership to track revenue trends, pipeline health, and product performance.

The long load times were frustrating stakeholders and causing **delays in decision-making**. Most users either stopped using them or relied on screenshots instead of real-time data. While technically “functional,” I believed this was **not acceptable performance for high-visibility tools**, and decided to raise the bar on performance and usability.

**A – Actions (~3–5 minutes)**

I started by **diagnosing the bottlenecks**:

1. **Live connections to large datasets**: Many dashboards were pulling data directly from live sources (e.g., Salesforce or Snowflake) with every user interaction, causing major lag.
2. **Inefficient SQL logic**: The underlying queries were not optimized. I noticed several bad practices:
   * Selecting \* instead of specific columns
   * No indexing or partitioning on large tables
   * Joining unnecessary lookup tables
   * Pulling multiple opportunity records instead of filtering to the latest one

Here’s how I addressed it:

* **Switched to Extracts**: I created Tableau data extracts that refreshed on a daily schedule, tailored to each dashboard’s refresh requirements. This instantly reduced backend query load.
* **Rewrote SQL Logic**:
  + Replaced SELECT \* with only required fields
  + Added proper **WHERE** clauses to reduce data volume
  + Used **window functions** like ROW\_NUMBER() OVER (PARTITION BY account ORDER BY opportunity\_date DESC) to get only the **most recent opportunity** per account
  + Eliminated unnecessary joins
* **Performance Testing**: Benchmarked performance before and after each change, using Tableau’s built-in performance recorder and SQL execution logs.
* **Stakeholder Training**: I documented the changes and trained the BI team to apply these best practices to other dashboards.

**R – Results (~1–2 minutes)**

After implementing these optimizations:

* Dashboard load times **dropped from 50–80 seconds to 10–15 seconds**
* Stakeholder usage increased significantly — users no longer had to wait or request offline reports
* Sales leadership cited improved responsiveness in QBR and weekly pipeline meetings
* These practices became **standard guidelines** within the BI team for all future Tableau development

**Amazon Leadership Principle: Insist on the Highest Standards**

**Story Title**: *Data Drift and Quality Check for Churnlytics Model Using Python*

**S/T – Situation and Task (~1 minute)**

At Equinix, we developed a **churn prediction model (Churnlytics)** to forecast whether customers were likely to discontinue services. This model used quarterly refreshed data and was critical for informing retention strategies.

Over time, I noticed that **our model’s churn predictions fluctuated heavily across quarters**, even though customer behavior hadn’t significantly changed. This inconsistency **eroded stakeholder trust** in the model outputs. I suspected that **data input quality was the root issue**, particularly input drift between quarters, but we had no mechanism to detect or measure it.

I took the initiative to design a reusable tool that could **automatically flag data drift and input inconsistencies**, aiming to enforce higher standards in our ML pipeline and improve downstream reliability.

**A – Actions (~3–5 minutes)**

I created a **Python-based data quality script** that could compare two datasets—typically the current and previous quarter’s model inputs—and detect structural and statistical drift.

Here’s what the tool did:

1. **Feature Drift Detection**:
   * Calculated drift percentages for each feature
   * Flagged categorical and numerical columns whose distributions changed significantly
   * Ranked features from highest to lowest drift using statistical thresholds
2. **Structural Checks**:
   * Identified **missing columns**, **unexpected nulls**, **duplicate rows**, and **unique identifier breakdowns**
   * Compared column-level statistics (mean, median, cardinality)
3. **Output Delivery**:
   * Generated a **text summary report** and a **CSV of flagged features**
   * Output connected to a Tableau dashboard where teams could visually inspect drift patterns over time
4. **Reusability & Collaboration**:
   * Parameterized the script to work with any two datasets
   * Shared it with the Data Science, Finance, and BI teams so it could be applied across models and quarterly loads
   * Documented usage with examples and published to the internal code repository

**R – Results (~1–2 minutes)**

The Python script **improved the reliability and interpretability** of the churn model significantly. Analysts were now able to **catch input issues before retraining the model**, preventing bad data from influencing results.

As a result:

* Model outputs became more **stable and explainable quarter-over-quarter**
* The BI and Data Science teams adopted the script as part of their **standard model input validation process**
* This tool was later repurposed by the Finance team for validating quarterly financial input files as well

By holding our data pipeline to a higher standard, we restored confidence in the Churnlytics model and improved the **overall data quality culture across departments**.

**Amazon Leadership Principle: Think Big**

**Story Title**: *Sales Performance Cockpit – Consolidating 15+ Dashboards into One Unified View*

**S/T – Situation and Task (~1 minute)**

At Equinix, our Data Analytics team managed over 20 active Tableau dashboards, each tracking a different aspect of business performance — including Sales, Bookings, New Logos, Funnel, and Customer Engagement. While each dashboard was valuable on its own, **stakeholders found it overwhelming and time-consuming to navigate** between so many disconnected views.

This created friction for executives who needed to see **end-to-end performance** across functions quickly, especially during QBRs and strategic reviews. I realized we needed a **radically different approach** — something more intuitive, consolidated, and scalable.

I proposed and led the creation of a **Sales Performance Cockpit (SPM)**: a single, interactive dashboard that provided a unified view of all key sales metrics.

**A – Actions (~3–5 minutes)**

I started by auditing our dashboard inventory and identifying 15 core dashboards used most frequently by sales leadership. I then:

1. **Designed a Consolidated Data Model**:
   * Integrated disparate data sources (Salesforce, booking systems, funnel databases) into a **single data warehouse model** that supported cross-dashboard KPIs
   * Created shared dimensions like account, region, product, and time, ensuring data consistency across metrics
2. **Built a Dashboard Cockpit Framework**:
   * Created 15 **mini “dashboard containers”** within one unified Tableau dashboard — each acting as a compact summary of a full dashboard
   * Each tile showed high-level KPIs and trends (e.g., YoY growth, regional performance)
   * Added **interactive filters** across the cockpit to filter by sales segment, IBX, or product
   * Included **deep-link buttons** to redirect users to full dashboards for more detailed exploration
3. **User Experience Focus**:
   * Designed with executive readability in mind — clear typography, color-coded status indicators, minimal clutter
   * Added tooltip explanations for each KPI so users could instantly understand definitions
4. **Rollout & Training**:
   * Partnered with Sales Ops and Regional Directors to conduct walkthrough sessions
   * Created user documentation and FAQ sheets to increase adoption
   * Integrated the cockpit into the executive BI portal and pinned it as a landing page for leadership users

**R – Results (~1–2 minutes)**

After launch, the **Sales Performance Cockpit became the most accessed dashboard** across the sales BI ecosystem within just one month.

It **streamlined stakeholder workflows**, allowing leadership to get a high-level overview in seconds and drill down only when needed. It also improved **cross-functional alignment**, since all metrics were now tied to a single, consistent data model.

Usage analytics showed:

* A **60% drop in time spent switching dashboards**
* A **30% increase in active engagement** from VP-level users
* Stakeholders reported faster decision-making during QBRs and weekly business reviews

**Amazon Leadership Principle: Think Big**

**Story Title**: *Churnlytics Model + Tableau Dashboard – Turning Predictive Insights into Actionable Strategy*

**S/T – Situation and Task (~1 minute)**

At Equinix, our team developed a **churn prediction model (Churnlytics)** to identify customers likely to leave within the next 12 months. While the model worked well, I noticed a recurring issue: **stakeholders, especially customer success and sales leaders, couldn’t interpret or act on the churn predictions** effectively. There was no visibility into **how many customers were flagged**, where they were located, or how churn was trending over time.

The model results were simply stored in the data warehouse, without any visual layer to help guide business strategy. I realized that in order to make this project impactful at scale, we needed a way to **transform predictions into business insights** — so I took the initiative to build a **Tableau dashboard** that could do exactly that.

**A – Actions (~3–5 minutes)**

I started by extracting the model outputs from our data warehouse and analyzed what information would be most useful for Sales and CS leaders. I then:

1. **Designed key KPIs** for churn insight:
   * Total predicted churns by quarter
   * Churn breakdown by country, sales segment, and IBX region
   * Churn rate trends over time
   * Filters by product, geography, customer type
   * Top contributing factors (from model output scores) where available
2. **Built an intuitive Tableau dashboard**:
   * Used heat maps, trend lines, and KPI tiles to make the insights quickly scannable
   * Structured the layout so VPs and regional leaders could get both high-level and segmented churn visibility in seconds
   * Created filters for time, region, and customer tier to help customize views
3. **Socialized with stakeholders**:
   * Conducted demos for Sales Ops, CS leaders, and account managers
   * Incorporated feedback to refine the UI and business logic
   * Connected the dashboard to refresh automatically with new model outputs each quarter
4. **Pushed to production**:
   * Published the dashboard to our enterprise Tableau server
   * Integrated it into the SalesOps portal and embedded links in regular business review decks

**R – Results (~1–2 minutes)**

Post-launch, the dashboard became **the primary tool used by Sales Ops and Customer Success teams to identify and act on churn risks**. Leaders used it during QBRs to focus on high-risk regions, target retention efforts, and proactively engage vulnerable customers.

* The dashboard had **over 80% adoption within 6 weeks**, and
* Stakeholders reported **faster churn investigation and better prioritization** of outreach efforts
* It also influenced the roadmap for retention programs, as regional leaders could now see *where* and *why* churn was happening more clearly

By bridging the model and the business, I helped ensure that a predictive tool became a **strategic asset**, not just a data science output.

**Amazon Leadership Principle: Bias for Action**

**Story Title**: *Last-Minute Territory Filter Addition for $30M+ Pipeline Review*

**S/T – Situation and Task (~1 minute)**

At Equinix, our team had just deployed a key **Sales Pipeline dashboard**, which was actively being used by Sales leaders to present insights to **VPs and directors**, covering over **$30M in opportunities**. The dashboard already included 8 standard filters such as region, segment, and product line.

Just before a major quarterly executive review, the sales team approached us with an urgent request: they needed the ability to **filter data by Territory and Sub-Territory** to align the narrative to how their teams were structured.

With the live dashboard already in production, and only **a few hours left before the presentation**, I had to move fast — but without compromising the stability or design integrity of the dashboard.

**A – Actions (~3–5 minutes)**

I immediately assessed the feasibility of the request and took the following actions:

1. **Investigated Source Fields**:
   * We didn’t have direct "Territory" or "Sub-Territory" fields in the pipeline dataset
   * I quickly collaborated with stakeholders and learned these could be derived from the **Account Owner metadata**
2. **Engineered the Fields On-the-Fly**:
   * Created **calculated fields** in Tableau to extract and standardize Territory and Sub-Territory values from account owner information
   * Ensured the mappings were accurate using lookups and validated sample outputs with stakeholders
3. **Designed a Non-Intrusive UI Addition**:
   * Instead of crowding the already full filter panel, I created an **“Additional Filters” toggle button** on the side of the dashboard
   * When clicked, it exposed the new Territory and Sub-Territory filters, keeping the main interface clean while giving access to advanced filtering
4. **Validated and Pushed to Production**:
   * Ran quick QA comparisons to ensure the new filters matched expected values by sales region
   * Pushed the update to production in time for the presentation
   * Stood by in case of live support needs — none occurred

**R – Results (~1–2 minutes)**

The dashboard was used successfully in the executive meeting just hours later — **with the new Territory-level insights displayed front and center**. The Sales team and executives appreciated the flexibility and precision the new filters enabled.

* The VP of Sales called out the update as a **“game-changer”** for aligning dashboards with how sales teams actually operated
* The additional filter toggle concept was so effective that we **standardized it across multiple dashboards** going forward
* This action not only met the immediate need, but also **introduced a UI pattern** that improved user control without clutter

**Amazon Leadership Principle: Bias for Action**

**Story Title**: *Rapid A/B Testing to Launch Improved Dashboard Visuals*

**S/T – Situation and Task (~1 minute)**

At Equinix, we had a critical internal Tableau dashboard used by senior sales and operations stakeholders. The original version was functional but suffered from poor alignment, inconsistent colors, and unclear layout, making it difficult to interpret key KPIs. Recognizing this could affect stakeholder trust and daily decision-making, I initiated a rapid visual redesign to improve clarity and user experience. However, we had no formal signoff from all teams yet, and waiting for approvals would delay improvements.

**A – Actions (~3–4 minutes)**

To move fast without risking disruption, I conducted a lightweight A/B test. I created a new version of the dashboard with a clean layout, consistent color palette, clearly defined sections (header, filters, KPIs, main body, and footer), and improved alignment.

Rather than pushing changes directly to all users, I split the stakeholder group into two test cohorts of around 50 people each. One group received the original dashboard, and the other received the new version. I tracked key engagement metrics using Tableau server logs—specifically time spent on the dashboard, frequency of return visits, and filter usage.

Additionally, I included an embedded feedback form for both groups to gather subjective input on layout clarity, ease of use, and perceived visual improvements. I coordinated directly with the enablement team and executive assistants to keep the experiment running for one full business week.

**R – Results (~1–2 minutes)**

The new version clearly outperformed the old one:

* **34% increase in average time-on-dashboard**
* **40% more interaction with filters**
* **Positive feedback from 88% of users in the new version group**, compared to 52% in the old group

Based on these results, we rolled out the improved version across the org. Stakeholder satisfaction improved, and the format became the **new design standard for future Tableau dashboards** in our BI team. This initiative showed how quick, calculated experimentation can lead to better outcomes without waiting for lengthy governance.

**Amazon Leadership Principle: Frugality**

**Story Title**: *QA Reporting Automation Using Python at Tata Elxsi*

**S/T – Situation and Task (~1 minute)**

At Tata Elxsi, I was part of the QA analytics team, where we reported key metrics weekly, including test case execution progress, pass/fail status, backlog items, and defect density. This was crucial for project managers and delivery leads to track QA effectiveness.

The reporting process involved downloading test data from our test management system, manually manipulating Excel files, applying formulas, and creating charts — which took **over 10 hours per week**. Our team didn’t have access to paid BI tools or automation platforms, so we needed a **cost-effective, scalable way to streamline this**.

**A – Actions (~3–4 minutes)**

I took the initiative to develop an end-to-end **Python automation pipeline** using entirely open-source tools. Here's what I did:

1. **Data Ingestion and Cleanup**:
   * Used pandas to read exported Excel reports
   * Cleaned and standardized test result columns across multiple sheets
2. **Metric Calculations**:
   * Automated KPIs like defect density, test pass/fail %, backlog count, execution coverage
   * Implemented dynamic filtering by week and module
3. **Visualization**:
   * Used matplotlib and seaborn to generate charts similar to what was manually created in Excel
   * Added auto-saving of the output graphs and summary reports as a PDF
4. **Reusable Codebase**:
   * Packaged the scripts into a reusable format
   * Documented usage so any teammate could generate the report by running one script
5. **No Cost, No Delay**:
   * Used only open-source tools, requiring **zero additional budget**
   * Shared the automation with other QA teams who had similar needs

**R – Results (~1–2 minutes)**

The Python automation reduced manual effort by **over 90%**, saving **10+ hours every week** for our team. Reports that previously took half a day were now generated in **under 3 minutes**.

* 📉 **Saved ~40 hours/month of manual effort**
* 💰 **Zero cost implementation** using Python + Pandas + Matplotlib
* 📊 **Improved report accuracy** by eliminating manual formula errors
* 🔄 Script reused across 3 QA teams within the organization
* 🏆 Recognized by my lead for solving a recurring issue without spending a rupee

**Amazon Leadership Principle: Frugality**

**Story Title**: *Standardization of Dashboard Design Across 15+ Tableau Dashboards*

**S/T – Situation and Task (~1 minute)**

At Equinix, our team managed over 15 business-critical Tableau dashboards, covering metrics such as revenue, new logos, churn, booking pipeline, and product attach rate. While each dashboard served its purpose, they were **inconsistently formatted** — filters were in different places, KPI cards varied in style, and header/footer structures lacked uniformity.

This inconsistency caused **stakeholder confusion**, repeated explanation requests, and inefficiency in navigation. We needed a **cost-effective** way to **streamline the user experience** without licensing new software or investing in UI consulting.

**A – Actions (~3–4 minutes)**

I initiated and led a project to **standardize all dashboard layouts using only native Tableau capabilities**. Here's how I approached it:

1. **Conducted a UX audit**:
   * Reviewed each existing dashboard for layout inconsistencies
   * Gathered feedback from end users and power stakeholders on what layout elements confused them the most
2. **Designed a reusable standard template**:
   * Created a Tableau template with a consistent structure:
     + **Header**: Dashboard title, last refresh timestamp
     + **Filter Panel**: Fixed position and consistent naming
     + **Body**: KPI blocks followed by trend visuals
     + **Footer**: Navigation links and tooltip help icon
   * Used Tableau containers, layout grids, and reusable workbook styles to avoid any third-party tools
3. **Socialized and got buy-in**:
   * Shared the template with key dashboard owners and BI leads
   * Incorporated feedback and finalized a version that met the needs of multiple teams
4. **Rolled out standardization**:
   * Refactored existing dashboards to match the template
   * Used “copy-paste” workbook components to save time and maintain visual consistency
   * Documented the template so future dashboard developers could use it too

**R – Results (~1–2 minutes)**

The standardized format significantly improved dashboard usability:

* 📈 **Dashboard engagement increased by 40%** (measured via Tableau usage logs)
* ❓ **Stakeholder support queries dropped by 50%**, indicating better self-service and understanding
* 💰 **Zero additional cost** — no new software, licenses, or design consultants were needed
* 🔁 The standardized format was **adopted as the default structure for all new dashboards** in our BI team
* 📄 Reduced development time for future dashboards by **up to 30%**

**Amazon Leadership Principle: Learn and Be Curious**

**Story Title**: *Learning PySpark to Scale Customer-Product Analysis for 30M+ Records*

**S/T – Situation and Task (~1 minute)**

While working at Equinix, I received a stakeholder request to analyze **customer behavior across all products** — essentially to identify what product combinations different customer segments were using.

The dataset was massive — over **30 million rows**, with detailed logs of customer interactions, product features, and segments. My usual tool, Pandas, wasn’t scaling — operations were slow, memory was maxing out, and it became nearly impossible to filter and transform the data effectively.

Rather than wait for someone else to step in or switch to a more limited sample, I decided to **learn PySpark**, knowing it could help me process data in-memory and in parallel — at scale.

**A – Actions (~3–4 minutes)**

I started by outlining exactly what I needed to learn, and took the following steps:

1. **Structured Learning Path**:
   * Took a fast-paced online course focused on PySpark for DataFrames and transformations
   * Studied Spark’s architecture: **RDDs**, **DataFrames**, and **lazy evaluation**
   * Practiced key concepts like **actions vs. transformations**, **caching**, and **partitioning**
2. **Hands-on Experimentation**:
   * Created a sandbox environment using Spark on my local machine and small datasets
   * Compared Pandas vs. PySpark for identical filtering and group-by tasks to build confidence
3. **Applied Learnings to Real Use Case**:
   * Imported the full 30M row dataset into HDFS and built an ETL pipeline using PySpark
   * Performed complex joins across product tables and segmentation logic using groupBy, agg, and filter
   * Used repartition() and cache() to optimize for iterative transformations
   * Wrote the cleaned and enriched output to our data warehouse for reporting
4. **Knowledge Sharing**:
   * Documented performance benchmarks: **Pandas script ran in 80+ minutes**, PySpark completed in under **15 minutes**
   * Created a reusable script template for future large-scale product-customer analysis

**R – Results (~1–2 minutes)**

The move to PySpark had a huge impact:

* 🚀 Reduced processing time by **over 80%** — from 80+ minutes in Pandas to under 15 minutes in PySpark
* 📈 Enabled analysis of the **full dataset** instead of a sampled subset, leading to better segmentation insights
* 🛠️ Reusable PySpark code adopted by teammates working on similar large datasets
* 📚 Earned informal recognition from my manager for being **“proactive in learning and scaling the team’s analytics”**

But most importantly, I developed a skill that made me more effective in handling large-scale data and set me up for similar challenges going forward.

**Amazon Leadership Principle: Learn and Be Curious**

**Story Title**: *Enhancing Dashboard Interactivity through Advanced Tableau Techniques*

**S/T – Situation and Task (~1 minute)**

At Equinix, I was responsible for maintaining several Tableau dashboards used by sales and operations stakeholders. One repeated concern was that **certain dashboards were too cluttered**, with too many visuals (e.g., booking by region, product, segment, etc.) displayed simultaneously, which made it **difficult to focus** and interpret insights effectively.

We needed to find a way to simplify the dashboards without losing any analytical depth. I realized this required techniques beyond what I was currently using, so I **took the initiative to upskill myself**.

**A – Actions (~3–4 minutes)**

To address this, I enrolled in an **advanced Tableau course on Coursera**, focused specifically on UI optimization and interactivity. I learned several key techniques, including:

* **Sheet swapping**: toggling between different visuals using user input
* **Dynamic visibility**: hiding/showing elements based on filter logic
* **Button actions and image-based navigation**
* **Performance tips** like using empty extracts to improve dashboard responsiveness

After gaining hands-on practice, I applied these to a dashboard that previously showed booking metrics across multiple dimensions in a single crowded view. Here’s what I did:

1. **Rebuilt the layout** to include a **navigation panel** with buttons that let users choose which KPI or breakdown (product, region, customer segment) to view
2. **Used sheet swapping logic** to dynamically load only the selected chart — reducing visual overload
3. **Integrated hover tooltips** and icons to guide users
4. **Tested responsiveness** using an empty extract setup to simulate real-time experience without full data loads

Finally, I demoed the revised dashboard to stakeholders and iterated based on feedback before pushing it to production.

**R – Results (~1–2 minutes)**

The redesigned dashboard had a **much cleaner interface**, and feedback from stakeholders was overwhelmingly positive:

* 📈 **User engagement increased by 50%** based on Tableau Server logs
* ❓ **Stakeholder questions dropped by 40%**, indicating improved clarity
* 🚀 Dashboard load time also improved due to more efficient extract usage
* 🧠 Internal BI team adopted similar interaction patterns for other dashboards moving forward

Additionally, I documented the techniques in our internal knowledge base to help teammates adopt these advanced features in their own projects — extending the value of what I learned.

**Amazon Leadership Principle: Earn Trust**

**Story Title**: *Supporting SQL and Statistics Students as a UTA Teaching Assistant*

**S/T – Situation and Task (~1 minute)**

While pursuing my master’s at the University of Texas at Arlington, I worked as a **Graduate Teaching Assistant** for the "Introduction to Data Science" course, which had over **120+ students**.

Many students were **struggling to grasp SQL syntax and logic**, especially around core clauses like SELECT, GROUP BY, and subqueries. They were overwhelmed by abstract explanations and often got stuck during assignments and exams. My responsibility was to support the professor with grading and student doubts, but I saw an opportunity to help students **understand the material more intuitively and build confidence**.

**A – Actions (~3–4 minutes)**

I went beyond routine duties and took the following steps:

1. **Identified the gaps**:
   * Analyzed common mistakes across SQL assignments
   * Noticed students struggled to visualize query logic (e.g., when to use HAVING vs WHERE)
2. **Designed a visual cheat sheet**:
   * Created a comprehensive infographic mapping the SQL execution order using color-coded blocks
   * Illustrated SELECT, WHERE, GROUP BY, ORDER BY, JOIN, and LIMIT with side-by-side visual and textual formats
   * Shared it across the class and received immediate positive feedback
3. **Held regular review sessions**:
   * Scheduled weekly **open Q&A hours** to walk through sample queries live
   * Used screen sharing and whiteboarding to visually deconstruct logic step-by-step
   * Created custom examples for difficult topics like nested subqueries and case statements
4. **Extended support based on student requests**:
   * After initial success, students requested visual explanations for **window functions**
   * Created a second cheat sheet for ROW\_NUMBER(), RANK(), LAG(), and LEAD() functions with clear use cases
5. **Maintained trust and availability**:
   * Responded promptly to student emails and office hour requests
   * Built a classroom culture where students felt comfortable asking questions

**R – Results (~1–2 minutes)**

The effort led to measurable and visible improvements:

* 📈 Many students reported their **understanding of SQL improved significantly**, with better assignment scores and less reliance on rework
* 🧠 **Exam scores improved by ~15%** on average in SQL-based questions compared to the first quiz
* 👏 Several students emailed the professor personally acknowledging how helpful the visuals and live sessions were
* 🧑‍🎓 I was personally commended by the professor and students for **building trust and going the extra mile**
* 🔁 The cheat sheets I created were used again in future semesters by the professor for new batches

This experience helped me realize the value of **empathetic communication, proactive support, and trust-building in any team setting** — whether academic or professional.

**Amazon Leadership Principle: Earn Trust**

**Story Title**: *Mentoring Data Analyst Interns to Ownership of Key Dashboards*

**S/T – Situation and Task (~1 minute)**

At Equinix, I was part of the BI team managing business-critical Tableau dashboards used by various sales and product stakeholders. During one summer, two new **Data Analyst interns** joined the team. They were bright and enthusiastic, but unfamiliar with our business context, KPIs, and tools like Tableau and our SQL-based data warehouse.

My manager asked me to **onboard and mentor them** so they could be productive and possibly take over ongoing maintenance of two dashboards. I saw this as a chance to not only support their learning but also free up team bandwidth by building their confidence.

**A – Actions (~3–4 minutes)**

I approached the mentoring effort with structure and empathy:

1. **Business and Data Deep Dive**:
   * Held a kickoff session explaining Equinix's core business — data centers, interconnection services, and revenue streams
   * Walked them through **key data domains**: bookings, product hierarchy, sales segments, and IBX region mappings
2. **Tool and Dashboard Training**:
   * Gave a hands-on walkthrough of **Tableau dashboards**, highlighting how we use filters, calculated fields, and parameter controls
   * Introduced them to our **data warehouse model**, including how booking and new logo tables were joined
3. **Task Ownership with Support**:
   * Assigned each intern one dashboard to shadow and eventually own
   * Paired with them during enhancement requests to model how we translate stakeholder asks into data actions
   * Reviewed their work, gave feedback on dashboard formatting, tooltips, and calculation logic
4. **Fostered Independence**:
   * Created short documentation for each dashboard with logic, owners, and refresh schedules
   * By the 6th week, they were able to troubleshoot refresh issues and respond to minor stakeholder questions themselves

**R – Results (~1–2 minutes)**

The mentorship effort paid off in multiple ways:

* 🎯 Both interns successfully **took over ownership of two live dashboards**, handling weekly updates and minor enhancements
* ✅ Stakeholders reported **no disruptions or data issues** after the transition
* ⏱️ I was able to save ~5–7 hours per week by delegating dashboard maintenance, allowing me to focus on strategic projects
* 👏 My manager appreciated the effort and cited my name during the end-of-internship review as someone who **"built trust and accelerated intern impact"**
* 🌱 One intern later returned for a full-time interview with Equinix and cited the mentorship experience as a highlight

**Amazon Leadership Principle: Dive Deep**

**Story Title**: *Fixing the Broken Intake Process with Smartsheet Tracking*

**S/T – Situation and Task (~1 minute)**

At Equinix, I was part of the BI team handling internal analytics (IA) requests. These included a mix of ad hoc analysis, Tableau enhancements, data model changes, and new dashboard builds.

However, our intake process was disorganized — stakeholders would make requests via Teams calls, emails, or in meetings, and we used a shared Excel sheet to track some of them. This led to multiple issues:

* Requests were **frequently missed or delayed**
* Stakeholders had **no visibility** into the status of their asks
* Our team had **no centralized accountability**

I wanted to dive deep into this recurring problem and build a more reliable intake workflow.

**A – Actions (~3–4 minutes)**

1. **Root Cause Analysis**:  
   I reviewed our past IA backlog and spoke with team members to map how requests were currently being received and recorded.
   * Over 40% of the requests were verbal or email-based with no proper follow-up
   * Excel tracking had no categorization, priority tagging, or ownership
2. **Solution Design – Smartsheet Tracker**:  
   I proposed a Smartsheet-based intake system with:
   * A structured **form** that stakeholders could fill out, including fields like request type, priority, owner, dashboard name, and due date
   * A **centralized dashboard** showing open, in-progress, and completed requests
   * **Automated reminders and status columns** to keep stakeholders informed
3. **Categories & Workflow Clarity**:  
   I added predefined request types (e.g., “New Dashboard”, “Add Field to Data Model”, “UI Change”) to help us triage work more efficiently and filter by type or team.
4. **Stakeholder Adoption**:  
   I demoed the form and tracker in a weekly sync, addressed concerns, and collaborated with our lead to make it a **required step** for any new IA submission.

**R – Results (~1–2 minutes)**

The impact was both immediate and lasting:

* 📉 **Missed requests dropped by over 80%** within the first two months
* 📈 Team productivity improved — we spent less time chasing down untracked requests
* 🔄 Stakeholders gained transparency into request status and timelines, reducing repeat follow-ups
* 💬 Our senior manager called it “a major operational win” and encouraged other teams to replicate it

This project taught me the value of **deep process thinking**, not just applying surface-level fixes — and how **structured intake systems** can transform stakeholder trust and delivery efficiency.

**Amazon Leadership Principle: Dive Deep**

**Story Title**: *Creating a Data Dictionary to Drive Clarity and Self-Service Across BI Models*

**S/T – Situation and Task (~1 minute)**

At Equinix, I worked on multiple Tableau dashboards and data models sourced from Salesforce, booking systems, and customer hierarchies. These models contained dozens of KPIs and calculated fields like MRR, NRR, booking buckets, and sales account types.

However, I quickly noticed a recurring challenge: **we lacked a centralized data dictionary.**  
New analysts (including myself when I joined) and stakeholders often asked, “What does this field mean?”, “Is this value pre-filtered?”, or “What’s the difference between booking and billing accounts?” — leading to confusion, repetitive clarifications, and slower onboarding.

I decided to take ownership and build a scalable solution to make this knowledge accessible and reduce these recurring questions.

**A – Actions (~3–4 minutes)**

1. **Identified the Gaps**:
   * I reviewed our data models and compiled all calculated fields and dimensions used in dashboards
   * Partnered with my manager and senior analysts to gather the **business logic and context** behind key KPIs
2. **Built a Structured Data Dictionary**:
   * Used **Smartsheet** to design a user-friendly dictionary with columns like:
     + Field Name
     + Source Table
     + Definition / Calculation Logic
     + Business Owner
     + Model it Appears In
   * Included examples and usage notes for frequently misunderstood metrics like Parent Name, MRR, Booking Buckets
3. **Made it Interactive and Accessible**:
   * Published the Smartsheet as a **filterable dashboard**, allowing users to select a data model (e.g., Salesforce Opportunities) and view only relevant fields
   * Enabled dropdown filters and hyperlinks to dashboards where each metric was used
4. **Promoted Adoption**:
   * Walked through the data dictionary during team syncs and onboarding sessions
   * Added the dictionary link to our dashboard headers and Confluence pages for easy access

**R – Results (~1–2 minutes)**

The data dictionary led to clear, measurable outcomes:

* 🧭 **Reduced repeat metric clarification questions by over 60%** within the first quarter
* 👩‍💼 New team members ramped up faster and understood dashboards without needing walkthroughs
* 📊 Analysts and product stakeholders began using it as a **self-service reference**, saving time for the BI team
* 💬 Feedback from stakeholders: *“This is like a Rosetta Stone for our data models — thank you!”*
* 🔁 Eventually, the same structure was used as a **template for other teams** (marketing analytics and finance)

This project taught me the power of **diving deep into root causes** — rather than answering the same question 20 times, I built a solution that answered it once and for all.

**Amazon Leadership Principle: Have Backbone; Disagree and Commit**

**Story Title**: *Adding Booking Targets to Dashboards Despite Initial Pushback*

**S/T – Situation and Task (~1 minute)**

At Equinix, our BI team developed a Tableau dashboard to visualize booking metrics across products, sales segments, and IBX regions. However, I noticed a major gap: the dashboard didn’t include **target or goal data**, which made it hard for sales leaders to interpret whether the actuals were on track.

Even though we had target values available in a separate source system, the team decided **not to include them**, citing concerns about data integration complexity and dashboard clutter.

I believed strongly that **without target context**, the dashboard was incomplete — especially from a sales executive’s perspective. So I raised my concern and proposed a solution.

**A – Actions (~3–4 minutes)**

1. **Challenged the Status Quo (with Respect)**:
   * During a team sync, I explained that **sales leaders need to know not just actuals, but performance vs. target**, and that we were under-serving our users
   * While others were hesitant due to effort and dashboard complexity, I emphasized how this feature could **drive better decision-making and improve executive adoption**
2. **Disagreed Respectfully, Then Took Initiative**:
   * When the team chose not to prioritize it, I offered to **prototype the integration myself**
   * I pulled the target data from the source system, cleaned and joined it with the existing booking data at the time + region + segment level
3. **Enhanced the Dashboard UX**:
   * Used **dual-axis visuals** to show bars for actual bookings and overlayed lines for targets
   * Added **tooltips and conditional formatting** to highlight where actuals were above/below targets
   * Ensured performance remained optimized by filtering and summarizing target data appropriately
4. **Committed to Team Goals**:
   * Even though I disagreed with the initial decision, I continued contributing to the rest of the dashboard as planned and stayed aligned with team deliverables
   * I shared the enhanced version as a **non-disruptive enhancement**, making it easy for the team to adopt without rework

**R – Results (~1–2 minutes)**

* 🎯 The enhanced dashboard was rolled out with target lines, and **adoption from the sales leadership team spiked**
* ✅ Executives found it far more actionable — now they could **see monthly and quarterly progress at a glance**
* 💬 Several leaders sent positive feedback like *“This is exactly what we were missing. Helps us hold our teams accountable.”*
* 👏 My manager acknowledged my initiative during our team retrospective and said this was a great example of **“pushing for what’s right with customer needs in mind.”**

This experience taught me how to **respectfully challenge consensus**, take ownership of solutions, and still stay committed to team goals — even when my initial suggestion wasn’t accepted.

**Amazon Leadership Principle: Have Backbone; Disagree and Commit**

**Story Title**: *Funnel Dashboard Simplification – Balancing Technical Accuracy with Stakeholder Needs*

**S/T – Situation and Task (~1 minute)**

At Equinix, I was tasked with building a sales **Funnel Dashboard** in Tableau to visualize deal progress through pipeline stages like Pipeline, Commit, Upside, and Closed Won. I designed a technically detailed dashboard that not only showed the metrics but also embedded **logic definitions, backend calculations, and data source references** to ensure transparency.

However, during a review session, my manager gave me candid feedback: while the data was accurate, the dashboard was **too technical for stakeholders** and lacked intuitive visuals. The implication was that **business users might not engage** with the dashboard, despite the depth I had built in.

**A – Actions (~3–4 minutes)**

1. **Pushed Back Respectfully**:
   * I explained why I had included logic layers—so that users would trust the numbers and understand the derivation
   * I emphasized that stakeholders often challenge numbers, and having visible logic can build credibility
2. **Listened and Aligned**:
   * My manager appreciated the accuracy, but highlighted that **ease of use and readability were more important for executive-level users**
   * I acknowledged the feedback and committed to improving the user experience without compromising data correctness
3. **Took Ownership of Redesign**:
   * I separated backend logic into a hidden documentation sheet and **refocused the dashboard on visual storytelling**
   * Reorganized views into:
     + **Live Funnel Snapshot**
     + **Quarterly/YoY Funnel Trend**
   * Used **bar charts for quarterly totals** and **line graphs for year-over-year growth**
   * Removed clutter and added **filter options by segment, region, and product**
4. **Committed to the Outcome**:
   * Even though it meant scrapping and reworking major sections, I committed fully to the new version
   * I also made sure the final version was delivered on time without affecting sprint deliverables

**R – Results (~1–2 minutes)**

* 📈 The revised dashboard was widely adopted by the Sales and FP&A teams
* 👍 Stakeholders commented that it was **“much more digestible and actionable”**, while still accurate
* 🧠 I learned how to **balance technical depth with visual simplicity**, and that **standing firm initially is okay as long as you adapt for the right outcome**
* 👏 My manager acknowledged that I showed maturity in pushing for correctness but then committing to the better user experience once consensus was reached

**Amazon Leadership Principle: Deliver Results**

**Story Title**: *Inbox Dashboard Automation to Improve Stakeholder Engagement*

**S/T – Situation and Task (~1 minute)**

At Equinix, we maintained over 15 Tableau dashboards covering critical business metrics—sales funnel, new logos, booking trends, and more. However, we noticed that many stakeholders weren’t regularly engaging with the dashboards. Due to their demanding schedules, they often **forgot to check**, even though the data was crucial for their decision-making.

There was no structured reminder system or personalization in dashboard access. I identified an opportunity to **automate dashboard delivery to their inboxes** in a way that was both scalable and filterable.

**A – Actions (~3–4 minutes)**

1. **Identified the Root Cause**:
   * Analyzed Tableau usage logs and confirmed that **dashboard engagement was sporadic**
   * Conducted informal interviews with stakeholders to learn they wanted a more **passive way to consume updates**
2. **Designed a Frugal & Scalable Solution**:
   * Used **Smartsheet** as a lightweight intake form where stakeholders could:
     + Select which dashboards they wanted
     + Choose filters (e.g., region, time range, segment)
     + Set frequency (daily, weekly, monthly)
   * This approach avoided building a new tool and used **existing software licenses**
3. **Built the Backend in Python**:
   * Created a Python script that would:
     + Pull stakeholder preferences from Smartsheet
     + Take dynamic screenshots of the filtered dashboards using Tableau APIs
     + Embed the dashboard image and a direct link in an email
     + Automatically send the email on the defined schedule
4. **Tested, Iterated, and Rolled Out**:
   * Piloted with 10 stakeholders across sales and finance
   * Addressed email rendering issues and updated the screenshot resolution logic
   * Officially rolled it out across the org as “Inbox Dashboard Delivery”

**R – Results (~1–2 minutes)**

* 📈 Within the first month, over **70% of our primary stakeholders subscribed** to the inbox automation
* 📬 Dashboard engagement rose significantly—**weekly usage improved by over 60%**
* 🧑‍💼 Stakeholders reported that the emailed snapshots made it easier to stay informed and **proactively spot trends**
* ✅ The solution was **low-cost**, **fully automated**, and scalable across teams
* 👏 This became a **best practice replicated by other BI teams globally** at Equinix

**Amazon Leadership Principle: Deliver Results**

**Story Title**: *Fixing Year-over-Year (YoY) Logic to Improve Dashboard Accuracy and Business Decision-Making*

**S/T – Situation and Task (~1 minute)**

While working at Equinix, I noticed a flaw in the **Year-over-Year (YoY) KPI calculations** displayed across several business-critical dashboards, including revenue, bookings, and funnel performance. The logic compared full-year totals of the current year to the previous year—regardless of how much of the current year had passed. This was **misleading during partial-year periods** and could negatively impact how sales and finance leaders interpreted performance trends.

Recognizing the risk of **incorrect executive decision-making**, I made it a priority to fix the logic across all impacted dashboards in a way that was scalable and trustworthy.

**A – Actions (~3–4 minutes)**

1. **Audited Existing Logic**
   * Reviewed YoY calculations across 10+ dashboards
   * Confirmed the issue: full-year vs. partial-year comparisons were inflating metrics early in the year
2. **Designed a Precise Fix**
   * Developed a dynamic logic that compared **year-to-date (YTD) values** only—ensuring fair and consistent comparisons
   * Implemented the fix using calculated fields in SQL and Tableau, with clear documentation
3. **Cross-Team Collaboration**
   * Aligned with data model owners, Tableau developers, and business leads to validate the corrected logic
   * Explained the reasoning clearly to leadership to gain quick buy-in
4. **Implemented with Zero Downtime**
   * Rolled out the fix across all dashboards using version control and validation tests
   * Added contextual tooltips and updated KPI footnotes to improve transparency for users

**R – Results (~1–2 minutes)**

* ✅ Implemented the fix in **10+ high-visibility dashboards** used by sales, marketing, and strategy teams
* 📉 Eliminated confusion caused by inflated YoY numbers—especially during Q1 and Q2 reviews
* 🧠 Increased stakeholder **trust** in dashboard metrics and improved credibility of the BI team
* 🚫 Reduced follow-up questions and manual rework during leadership presentations
* 📈 Ensured that future year comparisons are **accurate, scalable, and audit-ready**