



# Modernizing JDE on AWS

What It Takes, What It Costs, and How to Get It Right

[syntax.com](https://syntax.com)

ORACLE | Partner

# Overview: The Decision You're Actually Facing

If you're running JD Edwards (JDE) on-premises, you're likely approaching a decision point, not because JDE is going away (Oracle has committed to Premier Support for JD Edwards EnterpriseOne 9.2 through at least December 2037), but because the infrastructure underneath it is aging.

Hardware refresh cycles, data center lease renewals, and rising operational costs are forcing a conversation that many IT leaders have been deferring. Compounding factors—VMware licensing changes, accumulating technical debt, database licensing inefficiencies, and

the widening gap between your current JDE tools release and what Oracle is shipping—are accelerating that timeline.

## Is it time to move JDE to the cloud?

For most organizations, the answer is yes. The more important question is how—and whether you can do it without disrupting the business processes that run on your ERP every day.

While there are several cloud platforms capable of running JDE, this guide focuses on what is involved in moving JDE to AWS. It's written for CIOs and IT leaders who are currently or soon-to-be evaluating that decision.

The guide covers what JDE modernization on AWS looks like, what makes it complex, what it costs, what becomes possible once you're there, and how to get it right—starting with a first step that produces a defensible business case without overcommitting your organization.



# What Makes JDE Migration Different

Cloud migration is not a generic exercise. Moving JDE to AWS is fundamentally different from migrating a file server or a web application—and organizations that treat it as a standard lift-and-shift often find out why the hard way.

JDE environments at mid-market and enterprise companies carry years of accumulated complexity:

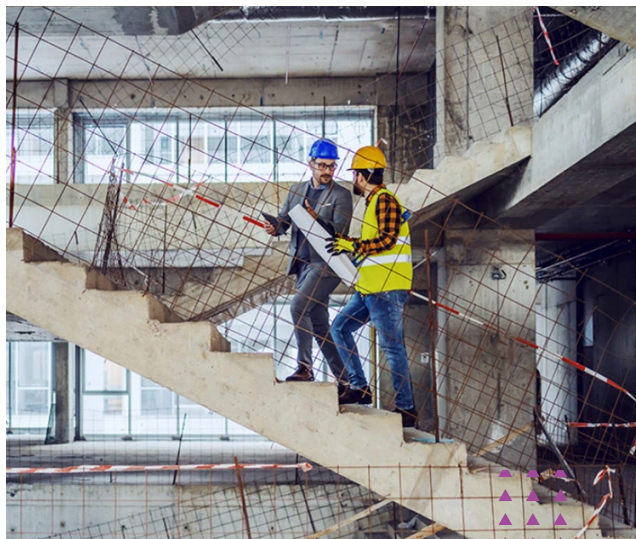
- Customizations built around specific business processes
- Integrations with third-party systems, such as EDI platforms, warehouse management solutions, shop floor systems, and tax engines that have to function without interruption the moment you go live
- Multi-entity, multi-currency, multi-language configurations that require careful sequencing
- A business that cannot tolerate downtime because operations run on JDE around the clock

Laticrete International is a good example of the complexity inherent in a long-time JDE environment.

Getting Laticrete's migration right required an ERP-first approach—understanding the application deeply before touching the infrastructure.

And that's why JDE knowledge matters when selecting a migration partner. Most cloud providers and many system integrators can move infrastructure competently. Far fewer understand JDE's unique architecture, upgrade path, customization patterns, and integration dependencies well enough to manage the migration without disrupting operations. The technical risk in a JDE migration lives in the application layer, not the cloud layer.

There are also compounding decisions that don't exist in simpler migrations. Do you upgrade JDE simultaneously with the migration, or sequence them separately? Combining both reduces long-term disruption but increases project complexity.



## Case-in-Point

Laticrete International, a global construction solutions manufacturer operating across 26 plants in over 100 countries, modernized JDE on AWS with Syntax. Their environment included 19 third-party integrations and a requirement for 24/7/365 availability across multiple time zones, languages, and currencies.

[Read the full Laticrete story](#) ►

Separating them is lower risk per phase but means two major change events instead of one. The right answer depends on your current release, your customization depth, your team's capacity, and your timeline—and it requires someone who understands JDE well enough to model both scenarios honestly.

[Learn about Syntax's JDE expertise](#) ►

# What JDE on AWS Actually Looks Like

A well-architected JDE environment on AWS is built for the same goals your on-premises environment was designed to meet—availability, performance, security, compliance—but without the capital burden of owning the infrastructure.

At the infrastructure level, Syntax deploys JDE on Amazon EC2 for compute, Amazon RDS or EC2-hosted databases for the data tier, Elastic Block Store (EBS) for primary storage, and Elastic Load Balancing for traffic distribution. Amazon S3 provides durable storage for snapshots, backups, and archival data. High availability is achieved through a multi-Availability Zone architecture aligned to the AWS Well-Architected Framework, distributing workloads across independent infrastructure so that a failure in one zone doesn't take down your ERP. Disaster recovery is designed from day one—not retrofitted after go-live.

The migration approach depends on your starting point. The primary options are:

- **Replatform (lift-and-shift):** Move JDE to AWS with minimal application changes. Fastest path, lowest migration risk, preserves your existing environment while eliminating the hardware burden. Best suited for organizations whose primary goal is infrastructure modernization and cost reduction on a defined timeline. For example, REC Solar took this path, moving their JDE environment to AWS largely as-is and achieving zero downtime post-migration.

- **Refactor (replatform + upgrade):** Combine the cloud migration with a JDE upgrade, which may also include a database or OS platform change. More complex, but eliminates two major change events, retires accumulated platform technical debt, and positions the organization to access current JDE features and tooling. This is the approach Syntax took with Laticrete—migrating from IBM i (AS/400) to native EC2 on Windows/SQL while simultaneously upgrading JDE and moving to AWS across a 26-plant global operation.

The realistic timeline for a well-scoped JDE migration to AWS is three to six months. That range reflects actual project experience. Environments with higher customization depth, more integrations, or more complex HA/DR requirements sit toward the higher end. Organizations that compress or skip the scoping phase consistently find the back half of the project expanding to compensate.

Post-migration, Syntax provides 24/7 managed services covering system monitoring, performance optimization, patching, and ongoing right-sizing—so your internal team isn't inheriting a new operational burden the moment the project closes.

**Learn about Syntax's AWS managed services**



## Case-in-Point

Trex Company, a global building products manufacturer running JDE on Windows/SQL, achieved a **recovery point objective (RPO) of under five minutes** and a **recovery time objective (RTO) of 15 minutes** into an alternate Availability Zone following their migration to AWS with Syntax.



# What Becomes Possible After Migration

Once JDE is running on AWS, the data that has been locked inside your ERP becomes accessible in ways that simply aren't possible on aging on-premises hardware.

The most immediate opportunity is automation. JDE processes that currently require manual intervention like purchase order approvals, work order updates, inventory adjustments, and intercompany transactions, can be automated through Microsoft Power Platform running against live ERP data.

Syntax has identified dozens of high-value automation opportunities specific to JDE customers that are fast to implement once the underlying infrastructure is cloud-ready. You can't build reliable automation on top of hardware you're managing out of a data center.

Beyond automation, AWS provides the data infrastructure that makes analytics and AI genuinely operational. Supply chain visibility, predictive maintenance signals, financial variance analysis, demand forecasting—these capabilities require real-time data pipelines, scalable compute, and modern tooling that cloud infrastructure enables (and on-premises environments constrain).

Syntax DnA<sup>3</sup>, our Digital Innovation Center of Excellence, is built on three pillars (AI, Automation, Analytics) and a Trusted Data Foundation. It's designed to help JDE customers build these capabilities progressively after migration, connecting ERP data to analytics and AI workflows without requiring a separate transformation program. None of this means the migration has to become a sprawling transformation program.

[See 6 quick-win JDE automations](#) ▶

[Explore Syntax DnA<sup>3</sup>](#) ▶



The migration itself is a contained, executable project with a defined scope and timeline. But organizations that approach it purely as an infrastructure cost exercise tend to underinvest in the architecture decisions that determine what they can do next.

Getting those decisions right from the start—data access patterns, integration architecture, security and governance—is what separates a migration that reduces costs from one that also expands capability.

# How to Make the Financial Case

The financial case for moving JDE to AWS has two components that need to be modeled separately: the ongoing cost of running JDE in the cloud (versus on-premises) and the cost of the migration itself.

On the ongoing cost side, the primary drivers are elimination of hardware refresh CAPEX, reduction of data center operational expenses, and the shift from unpredictable infrastructure costs to a predictable OPEX model. Organizations also reduce the internal labor burden of infrastructure management—patching, monitoring, capacity planning—which frees IT staff for higher-value work.

The magnitude of savings in your environment depends on your current hardware footprint, data center costs, staffing model, and licensing structure—which is why a TCO model built on your actual numbers is more useful than industry benchmarks. Syntax's Optimization and Licensing Assessment (OLA) is designed to build exactly this model, analyzing your existing Oracle licenses and infrastructure costs to produce a defensible financial picture before any migration commitment.

On the migration cost side, AWS's Migration Acceleration Program (MAP) is the most significant variable available to qualifying organizations. MAP is an AWS-administered program that provides funding, methodology, and tooling to offset migration costs—covering elements like the

**For qualifying Oracle application workloads, MAP can fund up to 65–75% of AWS migration spend as credits. That rises to 100% when the migration includes VMware modernization.**

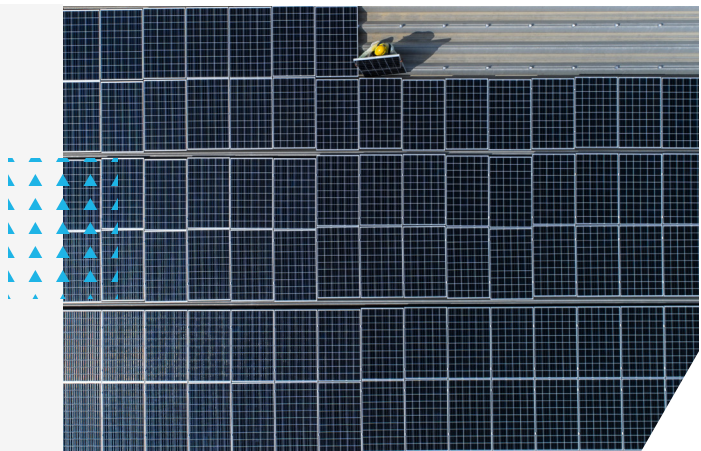
initial assessment, the landing zone build, first-wave migration execution, and modernization accelerators.

Syntax guides customers through the MAP process, from eligibility assessment through business case development and provider approval. Funding availability depends on environment size, executive sponsorship, timeline, and target platform, and is not guaranteed.

The combination of MAP funding eligibility and long-term operational savings means the financial case is often stronger than organizations expect going in. The variable that most often delays a decision isn't the economics. It's the absence of a credible, organization-specific model that a CFO can stand behind.

## Case-in-Point

REC Solar reduced their annual JDE operating budget by nearly 25% following migration to AWS with Syntax, with zero infrastructure downtime post go-live.





## How to Start Without Overcommitting

The most common mistake organizations make at this stage is treating the migration decision and the migration execution as a single event. They're not. The right first step is building a business case, not issuing a project kickoff.

Syntax structures this through two programs designed specifically for organizations at the evaluation stage:

- The **Optimization and Licensing Assessment (OLA)** goes a layer deeper on the financial side—analyzing your existing Oracle licenses, evaluating TCO, and identifying co-investment opportunities including MAP eligibility. The output is a recommendation set your CFO can engage with directly.
- The **MAP Assessment** is a scoping engagement where Syntax assesses your current JDE environment—architecture, customizations, integrations, licensing, HA/DR posture—and produces a clear picture of what a migration to

AWS would actually involve. This is the Assess phase of AWS's Migration Acceleration Program, and it surfaces the variables that determine timeline, complexity, and cost while giving you the foundation for a credible business case.

Neither program requires a migration commitment. Both produce artifacts—architecture assessment, TCO model, migration approach options—that are useful regardless of what you decide next. If the business case holds up, you have everything you need to move forward. If it doesn't, you've avoided a costly project built on assumptions.

If you're 12 to 24 months from a hardware refresh decision or data center renewal, now is the right time to run one of these engagements.

**Want to move forward?  
Schedule a MAP Assessment or OLA**





## What's Next

For most organizations running JDE on aging infrastructure, the move to AWS isn't a question of *if* but of *how well*. Done right, it's a contained project that eliminates the cost and unpredictability of owning hardware and unlocks the automation and analytics you don't have now. With a partner who understands JDE as deeply as the cloud, it becomes more than an infrastructure upgrade—it's the foundation for what your business does next.



Start your journey at:

- ▶ [syntax.com](https://syntax.com)
- ▶ [hello@syntax.com](mailto:hello@syntax.com)

## About Syntax

Syntax is a leading global technology solutions provider driving enterprise transformation in the cloud. We help organizations modernize mission-critical applications through AI-enabled innovation, industry-tailored outcomes, and end-to-end solutions across strategic advisory, implementation, and managed services. Through our Boutique @ Scale approach, we blend boutique level agility and care with the reach and resilience of a global enterprise. Supported by strategic partnerships with Oracle and other technology leaders, we empower customers to scale faster, work smarter, and build for what's next.



ORACLE | Partner