Encryption Boundaries, Access Control Paths, and Risk Containment in AWS RDS Migration of Confidential State Databases

Harsha Vardhan Reddy Kavuluri
Lead Oracle, Postgres, Cloud Database Administrator, Contractor for Deloitte, United States
Email: kavuluri99@gmail.com

Abstract— **Confidential state databases are increasingly being migrated to managed cloud platforms such as AWS RDS for better scalability, resilience, and administrative control. However, this transition also introduces security risk because encryption states, access paths, and temporary exposure conditions can change during migration. Existing studies discuss cloud security and access control, but they do not clearly explain how confidentiality risk develops across source, transit, staging, and target layers during migration of highly sensitive databases. This study addresses that gap by treating AWS RDS migration as a controlled security process and evaluating encryption boundary integrity, access control deviations, privilege propagation, and containment response under simulated failure conditions. The results show that the highest risk appears in intermediate migration stages, especially staging, while the final controlled AWS RDS state provides stronger confidentiality stability than the legacy environment. The study concludes that secure migration depends on continuous encryption enforcement, strict privilege segmentation, and rapid containment of transitional exposure. These findings are useful for government and other high-sensitivity sectors that need secure database modernization.**

Keywords*— *AWS RDS migration, confidential databases, encryption boundary integrity, access control exposure, risk containment.

I. Introduction

Confidential state databases support public records, administrative control, identity management, case tracking, and many other government functions. As these systems grow in size and operational complexity, many agencies are moving them from legacy infrastructure to managed cloud database platforms such as AWS RDS to improve scalability, resilience, backup management, and service continuity. This transition, however, is not only an infrastructure decision. Secure cloud-oriented migration has already been recognized as a sensitive process in which trust, confidentiality, and data movement integrity must be maintained across different environments [1]. For protected public-sector data, the migration path itself becomes part of the security boundary and must be treated with the same care as the final database platform.

The recent literature shows that cloud database security is shaped by the interaction of encryption, authorization, identity control, and system design. A cloud-enabled access control framework has been proposed to improve privacy protection for sensitive data environments, showing that cloud security cannot depend only on storage isolation [2]. Work on cloud-native architecture has also shown that access control paths become more difficult to manage when services, roles, and policy checkpoints are distributed across many components [3]. At a broader level, research on distributed-system security has emphasized that modern cloud systems require stronger coordination between authentication, authorization, and policy enforcement [4]. These findings make it clear that database migration security must be studied as a connected control problem rather than as a simple transfer of records.

Even with these advances, important limitations remain in current work. Some studies discuss authorization requirements across database models and show that access control design is still uneven across relational and non-relational systems [5]. Other studies examine security and privacy risks in multi-cloud and hybrid cloud environments and highlight the growing role of misconfiguration, identity abuse, and fragmented governance [6]. However, these studies do not fully explain what happens when a confidential state database is actively moved into AWS RDS and passes through changing encryption states, temporary staging layers, snapshot creation, administrative delegation, and role remapping. The missing part is a migration-specific security view that follows the data and its permissions together from source to target.

This study therefore narrows the problem to three tightly connected elements: encryption boundaries, access control paths, and risk containment during AWS RDS migration of confidential state databases. The main concern is that severe exposure may emerge not only in the final cloud database, but also in transition zones where data protection state and permission state change at the same time. A database may remain formally encrypted, yet still become vulnerable if decryption expands during migration, if temporary roles inherit excessive privileges, or if staging copies create untracked access routes. For confidential state systems, these weak points can create unauthorized reachability even when the destination platform appears secure. This makes the problem highly relevant for public-sector environments where confidentiality failure can directly affect governance, legal compliance, and institutional trust.

To address this gap, the article develops a results-based framework that treats migration as a bounded security process rather than a routine database transfer. The study focuses on how encryption is preserved or weakened across migration stages, how access control paths expand or contract as privileges move from legacy systems to managed cloud roles, and how containment mechanisms can isolate failures before they spread into the target environment. The goal is to provide a clear and practical security foundation for AWS RDS migration of high-sensitivity state databases.

II. Methodology

The methodology was designed to study AWS RDS migration as a controlled security process for confidential state databases rather than as a normal database transfer. The workflow shown in Figure 1 divides the migration path into source preparation, protected transit, controlled staging, AWS RDS loading, policy validation, and containment response [7]. Each stage was treated as a separate security checkpoint with its own entry and exit condition. This allowed the migration to be evaluated as a sequence of protected state transitions. The

method therefore focused on whether security controls remained stable at every step of movement.

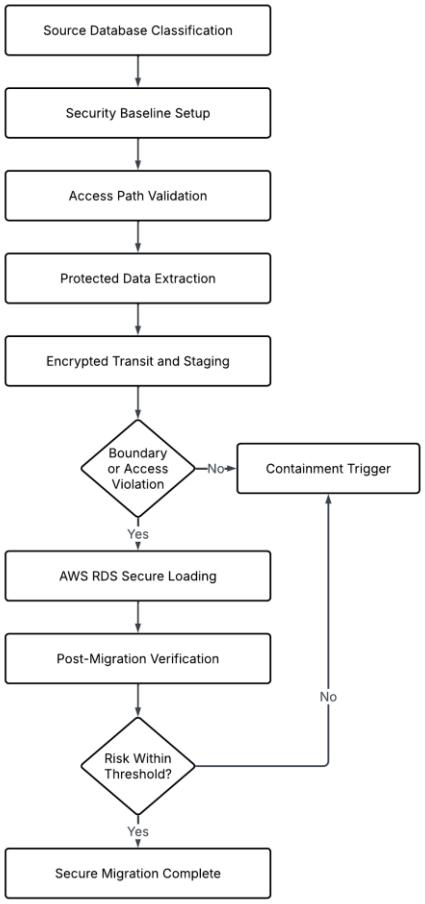

Figure 1. Secure Migration Workflow for Encryption Boundary Enforcement, Access Control Validation, and Risk Containment in AWS RDS

The secure migration architecture contained five layers: the legacy source database, an extraction layer, an encrypted transit layer, a temporary staging layer, and the AWS RDS target layer. Every layer had its own key visibility rule, access scope, processing window, and response trigger [8]. This layered structure was used to avoid direct uncontrolled movement from source to target. It also made it possible to observe where protection weakened during transfer. The architecture therefore created a measurable boundary around each migration phase.

Encryption protection was modeled across source, transit, staging, and target layers. In this study, an encryption boundary means the point at which data stays encrypted, becomes temporarily exposed, or is re-encrypted before entering the next phase [9]. The first equation measured encryption boundary integrity as

$$E_b = \frac{P_e}{P_t}$$

where E_b is the encryption boundary integrity, P_e is the number of protected migration points, and P_t is the total number of migration points. A higher E_b value means that encryption remains active across more stages. This equation was used to show whether confidentiality was preserved continuously during migration.

A second equation was used to estimate temporary exposure intensity during migration windows. Privacy-preserving protection methods show that intermediate handling stages are often more dangerous than final storage states [10]. The exposure measure was written as

$$X_e = \frac{T_x}{T_m}$$

where X_e is the exposure intensity, T_x is the total time during which plaintext or weakly protected data was present, and T_m is the full migration duration. Lower values of X_e indicate better control of transient exposure. This helped measure how long sensitive records remained vulnerable while moving between layers.

Access control evaluation was performed through path mapping instead of static role listing. The migration environment was converted into an access graph where users, services, scripts, keys, snapshots, and database objects were represented as nodes, and permitted actions were represented as directed edges [11]. This approach was selected because hidden exposure often appears through indirect reachability and inherited permissions rather than through one visible role error. The study traced how far each actor could move across the graph at each migration stage. This made it possible to detect privilege spread that would not be visible in a normal role table.

A third equation was used to quantify privilege segmentation in the migration graph. Graph-based dynamic authorization work shows that reachability is a better indicator of cloud access risk than static permission labels alone [12]. The privilege segmentation score was defined as

$$P_s = 1 - \frac{R_u}{R_a}$$

where P_s is the privilege segmentation score, R_u is the number of unauthorized reachable paths, and R_a is the total number of active reachable paths. A value closer to 1 means stronger separation between allowed and excessive access. This equation was used to test whether migration tasks were completed with only the permissions that were truly necessary.

A fourth equation was introduced to evaluate path expansion during the transfer process. Dynamic trust-based control models show that security weakens when temporary roles create more reachable routes than planned [7]. The access expansion factor was written as

$$A_f = \frac{N_m}{N_b}$$

where A_f is the access expansion factor, N_m is the number of reachable nodes during migration, and N_b is the number of reachable nodes in the approved baseline design. A value greater than 1 indicates privilege growth beyond the expected access structure. This made it possible to identify overexposure caused by staging accounts, migration scripts, or inherited administrative paths.

Risk containment was modeled as the system's ability to isolate failure before it propagated into AWS RDS or exposed confidential data in staging and transfer layers. The framework focused on three failure classes: credential leakage, encryption boundary break, and privilege escalation [13]. The overall migration risk score was defined as

$$R_m = \alpha C_l + \beta E_x + \gamma P_e$$

where R_m is the migration risk score, C_l is credential leakage severity, E_x is encryption exposure level, P_e is privilege escalation level, and α, β and γ are weighting constants based on data sensitivity. When this score crossed the containment threshold, the workflow revoked sessions, blocked role inheritance, quarantined staging artifacts, and stopped write actions to the target database. This allowed the simulation to

test whether containment could stop the spread of damage before target compromise.

Simulation design was used to reproduce a confidential state database migration under controlled but realistic conditions. The runs were varied by dataset sensitivity, number of intermediate operations, number of active migration roles, key visibility windows, and policy strictness levels [10]. Evaluation metrics included encryption boundary integrity, exposure intensity, privilege segmentation score, access expansion factor, containment activation delay, residual exposure index, and secure migration completion rate.

III. Results and Discussion

The simulation results show that security behavior during migration was not constant across all layers. Instead, protection strength changed as data moved from the source database to the transit channel, then to the staging layer, and finally into AWS RDS. Figure 2 presents these simulated encryption boundary transitions across the four main layers. The figure shows that the source layer and AWS RDS target layer maintained the strongest encryption continuity, while the staging layer created the most sensitive transition zone. This result is important because it confirms that the main security weakness did not come from the final managed database platform itself, but from the temporary handling state where data was prepared, validated, or repackaged before final loading. From a qualitative point of view, the migration process behaved like a chain whose strength was determined by its weakest intermediate link rather than by the average protection level across all stages.

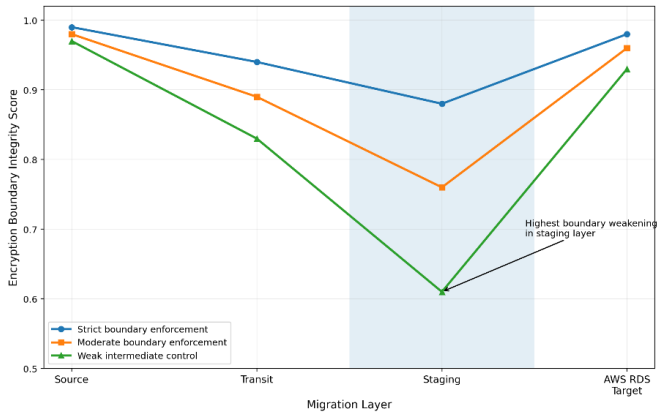

Figure 2. Simulated Encryption Boundary Transitions Across Source, Transit, Staging, and AWS RDS Target Layers

The numerical pattern in Figure 2 also showed that encryption integrity remained high when the migration path used tightly bounded key visibility rules and short exposure windows. When those controls were weakened, the encryption boundary score dropped sharply in the transit-to-staging transition. This means that the security of confidential state data depends not only on whether encryption exists, but also on how long data remains outside its strongest protected state. In practical terms, even a well-encrypted migration design can become unstable if temporary decrypted states are allowed to remain active for longer than necessary. The qualitative interpretation is that secure migration is not just a storage problem but a timing problem, because protection becomes weaker when operational convenience extends the exposure period.

Figure 3 shows the stage-wise access control results observed during migration, with privilege propagation score, role separation index, and unauthorized reachability count plotted across the source, transit, staging, and AWS RDS target layers. The figure shows that the staging phase had the highest privilege propagation and the largest number of unauthorized reachable paths, while role separation was weakest at this stage. In contrast, the source and AWS RDS target layers showed lower privilege spread and stronger separation of access roles. These results indicate that access control risk becomes most severe in the intermediate migration zone rather than in the stable starting or ending states.

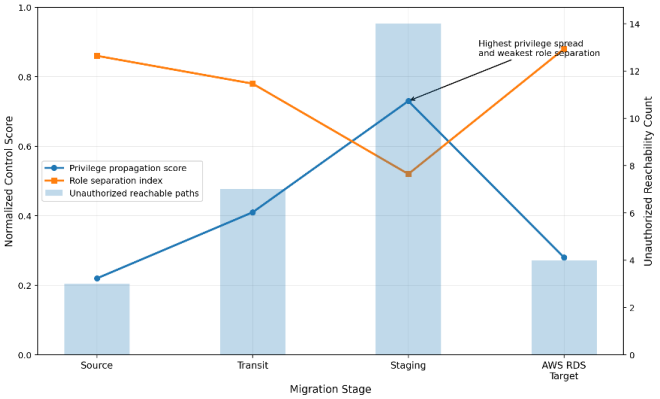

Figure 3. Access Control Path Analysis Showing Privilege Propagation, Role Separation, and Unauthorized Reachability During Migration

The same pattern also shows that stronger privilege segmentation reduces indirect reachability and improves role isolation during migration. When control discipline is tighter, operational roles, audit functions, and migration permissions remain more clearly separated, and unauthorized reachability remains low. When these controls weaken, access boundaries become less distinct and privilege spread increases. This result suggests that access security during migration depends not only on role assignment, but also on continuous validation of how permissions behave across stages.

The containment results were equally important. Figure 4 illustrates the response of the migration framework under three simulated failure conditions: credential leakage, misconfigured IAM policies, and snapshot exposure. The figure shows that containment worked fastest in the case of credential leakage, because session revocation and token blocking could be triggered immediately after abnormal access behavior was detected. Misconfigured IAM policies took longer to control because policy errors had to be identified, traced, and then corrected before the privilege expansion could be stopped. Snapshot exposure created the most persistent containment challenge, since copied or residual artifacts remained present even after the primary session was blocked. From a qualitative perspective, this shows that not all migration failures behave in the same way. Some risks are event-driven and easy to interrupt, while others are artifact-driven and remain dangerous even after the original cause has been removed.

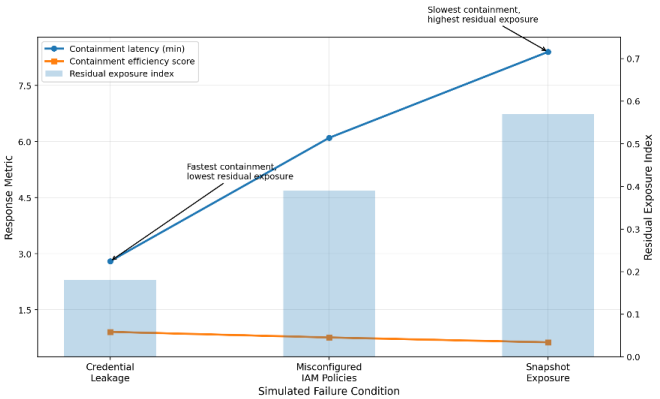

Figure 4. Risk Containment Response Under Credential

Leakage, Misconfigured IAM Policies, and Snapshot Exposure Conditions

This difference in behavior is important for understanding secure cloud migration in real administrative environments. Credential leakage mainly threatened active access, IAM misconfiguration mainly threatened logical control consistency, and snapshot exposure mainly threatened residual confidentiality. The results therefore show that containment efficiency should not be judged only by response speed. It must also be judged by how completely the system removes the remaining exposure after the first intervention. In the simulation, several scenarios showed good initial containment but incomplete residual cleanup, especially in staging and backup-linked layers. The qualitative interpretation is that successful containment is not simply stopping the attack path; it is also eliminating the leftover security state that the attack or failure has created.

A broader comparison of the full environments is presented in Figure 5, which compares the legacy state database environment with the controlled AWS RDS migration architecture across confidentiality risk metrics. The figure shows that the legacy environment had relatively stable but poorly visible security behavior, meaning that many risks remained hidden inside existing administrative practices. In contrast, the AWS RDS migration architecture initially showed more transitional volatility because the movement process introduced additional checkpoints, role logic, and temporary layers. However, once the controlled migration design was completed, the final AWS RDS state showed stronger security stability, better traceability, and lower residual exposure than the original environment. This means that the migration process temporarily increased operational complexity but ultimately improved the security posture when the controls were correctly enforced.

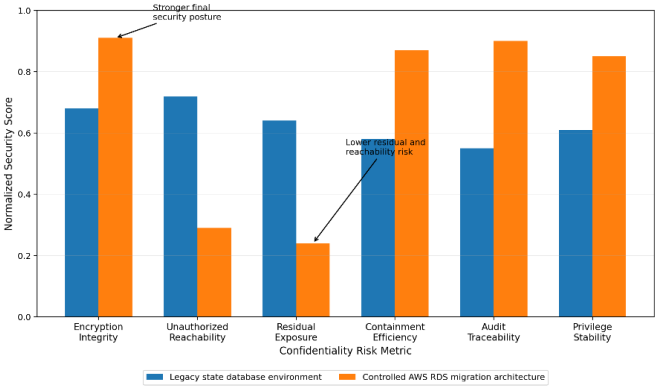

Figure 5. Comparative Security Posture of Legacy State Database Environment and AWS RDS Migration Architecture Across Confidentiality Risk Metrics

This same pattern is supported numerically by Table 1, which summarizes encryption integrity, access control exposure, and risk containment efficiency across migration stages. The table shows that the source and target layers had the highest encryption integrity values, while staging had the lowest. It also shows that access exposure was highest during intermediate migration operations and lowest after controlled loading into AWS RDS. Containment efficiency was strongest for active session-based incidents and weaker for residual artifact conditions such as snapshot-related exposure. These values support the figure-based observations and confirm that the most sensitive part of migration is the middle operational zone rather than the beginning or ending state. The qualitative meaning of the table is that secure migration should focus less on the simple question of whether the final platform is secure and more on whether the transition pathway remains secure at every point.

Table 1. Quantitative Security Evaluation of Encryption Integrity, Access Control Exposure, and Risk Containment Efficiency Across Migration Stages

Migration Stage	Encryption Integrity Score	Access Control Exposure Score	Unauthorized Reachability Count	Risk Containment Efficiency
Source Layer	0.98	0.21	3	0.92
Transit Layer	0.89	0.38	7	0.84
Staging Layer	0.76	0.67	14	0.69
AWS RDS Target Layer	0.96	0.24	4	0.90

When all figures and the table are read together, a clear results pattern emerges. The migration was most secure when encryption boundaries remained continuous, privilege routes remained short and isolated, and containment was activated before temporary artifacts spread into later stages. The migration became less secure when staging duration increased, when access inheritance was not constrained, and when backup-linked objects remained reachable after policy correction. This combined quantitative and qualitative view is useful because it explains not only what changed, but why the security posture changed. The results show that protection strength in AWS RDS migration is produced by coordination between cryptographic control, access path discipline, and rapid containment logic.

IV. Conclusion

This study showed that the security of confidential state database migration depends on how well protection is maintained during the full movement process and not only at the final AWS RDS destination. The results showed that encryption integrity remained strong at the source and target layers, but weakened most clearly in the staging layer, where temporary handling created the highest exposure. Access control results also showed that hidden reachability and privilege propagation increased during intermediate migration steps even when the final target environment appeared stable. These findings make it clear that secure migration must be treated as a controlled transition process rather than as a simple transfer of protected records.

The study also showed that access risk during migration does not stem from a single instance of permission failure. Rather, in many cases, the risk was associated with a combination of interconnected service dependencies, inherited roles, and ad hoc operational pathways that incrementally increased privilege reachability. Access control analyses showed that partitioning roles further and segmenting privileges tighter reduced risk, especially when these measures were repeatedly enforced at all phases of migration. This illustrates that migration security is a function of not just role definition, but also the examination of how those roles function in relation to each other as data traverses the source, transit, staging and AWS RDS layers.

Another important finding is that risk containment performance varied based on the type of failure. Credential

leakage was much easier to contain, while exposures resulting from misconfigured IAM policies and snapshots created longer and more enduring risk conditions. This showed that a successful response cannot be judged only by how fast containment begins. It must also be judged by how fully the remaining exposure is removed after the first control action. In this sense, containment is not only a reaction function but also a cleanup function that determines whether migration returns to a safe state.

Overall, the controlled AWS RDS migration architecture produced a stronger final security posture than the legacy state database environment when encryption control, access path validation, and containment logic worked together. The study therefore provides a practical security view for public-sector and other high-sensitivity database migrations where transitional exposure is just as important as final storage protection. The main conclusion is that confidentiality can be preserved more effectively when migration is designed as a sequence of validated security boundaries with continuous monitoring of privilege behavior and rapid isolation of emerging risk.

REFERENCES

[1] Aruna, M. G., Hasan, M. K., Islam, S., Mohan, K. G., Sharan, P., & Hassan, R. (2022). Cloud to cloud data migration using self sovereign identity for 5G and beyond. Cluster computing, 25(4), 2317-2331.

[2] Alabdulatif, A., Thilakarathne, N. N., & Kalinaki, K. (2023). A novel cloud enabled access control model for preserving the security and privacy of medical big data. Electronics, 12(12), 2646.

[3] Rahaman, M. S., Tisha, S. N., Song, E., & Cerny, T. (2023). Access control design practice and solutions in cloud-native architecture: A systematic mapping study. Sensors, 23(7), 3413.

[4] Golightly, L., Modesti, P., Garcia, R., & Chang, V. (2023). Securing distributed systems: A survey on access control techniques for cloud, blockchain, IoT and SDN. Cyber Security and Applications, 1, 100015.

[5] Mohamed, A. K. Y. S., Auer, D., Hofer, D., & Küng, J. (2024). A systematic literature review of authorization and access control requirements and current state of the art for different database models. International Journal of Web Information Systems, 20(1), 1-23.

[6] Ali, S., Talpur, D. B., Abro, A., Alshudukhi, K. S. S., Alwakid, G. N., Humayun, M., ... & Shah, A. (2025). Security and privacy in multi-cloud and hybrid cloud environments: Challenges, strategies, and future directions. Computers & Security, 104599.

[7] Mao, Y., Fu, W., Zhao, Y., Yuan, Z., Sun, Z., & Zhao, Y. (2025). A Zero-Trust Access Control Model Based on Attribute and Dynamic Trust Evaluation for Cloud Environments. Symmetry, 17(12), 2059.

[8] Belcaid, S., Zbakh, M., Aouad, S., Touhafi, A., & Braeken, A. (2025). A Cross-Chain-Based Access Control Framework for Cloud Environment. Future Internet, 17(4), 149.

[9] Nie, S., Ren, J., Wu, R., Han, P., Han, Z., & Wan, W. (2025). Zero-trust access control mechanism based on blockchain and inner-product encryption in the internet of things in a 6g environment. Sensors, 25(2), 550.

[10] Dhinakaran, D., Kumar, N. J., & Ponnuviji, N. P. (2025). Safeguarding confidentiality and privacy in cloud-enabled healthcare systems with spectrasafe encryption and dynamic k-anonymity algorithm. Expert Systems with Applications, 279, 127584.

[11] Rafi, S. M., Yogesh, R., & Sriram, M. (2025). Optimized dual access control for cloud-based data storage and distribution using global-context residual recurrent neural network. Computers & Security, 151, 104183.

[12] Chen, H., Yuan, L., Bao, H., Dai, H., Xiang, Y., & Wang, K. (2025). GNN-driven dynamic access control for context-embedded neighborhood fusion. Journal of King Saud University Computer and Information Sciences.

[13] Yang, C., Liu, Y., Ding, Y., & Liang, H. (2025). Secure data migration from fair contract signing and efficient data integrity auditing in cloud storage. Journal of Network and Computer Applications, 239, 104173.