AWS Service Accelerator: RedShift

Date: 2019-04-08

The Service Accelerator document is meant to provide a concise presentation of information to help organizations learn specific security patterns that can be implemented as part of their service configuration. Since many security programs follow the same security domains, this format provides a simple way to share the information and expedite the approval process for AWS services. The process to create this document involves a deep dive review of how the service works through hands on experience, documentation review, and discussion with AWS service teams.

The information in the Detailed Security Configuration section is meant to answer configuration/implementation questions from an operational perspective. Organizations can use the specific guidance in this document to create codified controls that meet their specific control requirements.

## 

Detailed Security Configuration

Overview

This section is meant to provide an opinionated approach towards the implementation security controls by domain. Although the approaches may not be a fit for all use casees or complete for production use, they are meant to guide the audience towards current best practices and design considerations to achieve security control objectives.

**Controls and Architectures**

This table maps Security Domain to the corresponding controls and architectural best practices as documented in AWS’ public documentation, white papers, and blog posts.

|  |  |  |
| --- | --- | --- |
| **Security Domain** | **Control & Architectural Suggestions** | **References** |
| **Encryption** |  |  |
| Encryption of data at-rest | AWS RedShift supports KMS and HSM to provide key material management and encryption services. Encryption at rest of RedShift encrypts the data blocks and system metadata of the cluster and its snapshots.[1]  **Note:** By default, Amazon Redshift selects the account service default key as the master key. The default key is an AWS-managed key that is created for your AWS account to use in Amazon Redshift. Some customer security controls prevent the use of default service KMS keys for sensitive workloads. Users should pre-create a customer managed CMK for RedShift usage.  **Implementation Note:** Encryption is an optional, immutable setting of a cluster. If you want encryption, you enable it during the cluster launch process. To go from an unencrypted cluster to an encrypted cluster or the other way around, unload your data from the existing cluster and reload it in a new cluster with the chosen encryption setting. [2]  For S3 encryption details see S3 Accelerator.  For KMS details see KMS Accelerator | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-db-encryption.html> 2. [Migrating an Unencrypted Cluster to an Encrypted Cluster](https://docs.aws.amazon.com/redshift/latest/mgmt/migrating-to-an-encrypted-cluster.html) 3. <https://docs.aws.amazon.com/cli/latest/reference/redshift/describe-clusters.html> |
| Encryption of data in-transit | * To support SSL connections, Amazon Redshift creates and installs an [AWS Certificate Manager (ACM)](https://aws.amazon.com/certificate-manager/) issued SSL certificate on each cluster. The set of Certificate Authorities that you must trust in order to properly support SSL connections can be found at <https://s3.amazonaws.com/redshift-downloads/redshift-ca-bundle.crt>.   Note: Customers can import certificates into AWS ACM to use custom certs and still take advantage of the integration ACM has with Redshift.[3]   * RedShift endpoints are available over HTTPS at a selection of regions.   Best practice:   * Set the **“require\_SSL”** parameter to **“true”** in the parameter group that is associated with the cluster. * For workloads that require FIPS-140-2 SSL compliance an additional step is required to set parameter **“use\_fips\_ssl”** to **“true”** | 1. How to encrypt end to end: <https://aws.amazon.com/blogs/big-data/encrypt-your-amazon-redshift-loads-with-amazon-s3-and-aws-kms/> 2. To make client side encryption work follow this pattern <https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html> 3. https://docs.aws.amazon.com/acm/latest/userguide/import-certificate.html |
| Encryption Key Management | * Key management is handled by KMS or HSM. * For encryption functions to occur in Redshift the DEK/CEK are stored on disk in an encrypted state. They persist on disk after cluster reboots and then require a request to KMS to use the CMK to decrypt the CEK to be used again in memory. * Key Rotation can occur as often as data requirement define.[1]   Example: Commandline key rotation[3]  rotate-encryption-key  --cluster-identifier <value>  [--cli-input-json <value>]  [--generate-cli-skeleton <value>]  **Note:** Snapshots stored in S3 will need to be decrypted prior to key rotation and then re-encrypted using the new DEK. This is a process that should be tested prior to production use. | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-db-encryption.html> 2. <https://docs.aws.amazon.com/kms/latest/developerguide/concepts.htm> 3. <https://docs.aws.amazon.com/cli/latest/reference/redshift/rotate-encryption-key.html> |
| **Infrastructure** |  |  |
| Isolation of physical hosts | **N/A**: RedShift is a fully-managed service and for cluster nodes the isolation of hosts is not currently possible via dedicated host ec2-resources. Reserved instances can be purchased to ensure availability of ec2 resource types. |  |
| Network Isolation | When an Amazon Redshift cluster is provisioned, it is locked down by default so nobody has access to it except to IAM entities with Console access from within the provisioned network and with the default credentials. Amazon Redshift provides a cluster security group called default, which is created automatically when you launch your first cluster. Initially, this cluster security group is empty. You can add inbound access rules to the default cluster security group and then associate it with your Amazon Redshift cluster. To grant other users inbound access to an Amazon Redshift cluster, you associate the cluster with a security group. To grant access use an existing Amazon VPC security group or define a new one and then associate it with a cluster. For more information on managing a cluster on the EC2-VPC platform, see [Managing Clusters in an Amazon Virtual Private Cloud (VPC)](https://docs.aws.amazon.com/redshift/latest/mgmt/managing-clusters-vpc.html).  Amazon RedShift relies on EC2 security groups to provide infrastructure security, and thus initial protection from unauthorized traffic connecting to the cluster. [1]  Best Practice   * SecurityGroups should follow a naming convention for the entire account * The cluster leader node is the only EC2 instance that is allowed to communicate with the cluster nodes in the AWS Service Account. Ensure to enable VPC FlowLogs on the leader node ENI and capture logs from the OS to ensure only authorized access and activity has occurred. * For the leader node it is recommended the SecurityGroup have no outbound entries to prevent egress of data if the node is compromised. * Attempt to reference other SecurityGroups instead of using IP * Enable EnhancedVPCRouting [4] * Enable VPCEndpoint with S3 * For most use cases the cluster should not be publically accessible. * Configure default port to authorized port for SecurityGroup usage. The default port is 5439 and cannot be changed after cluster is built.[5]   See S3 Accelerator for controls around S3 and data isolation. | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-security-groups.html> 2. <https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-security-groups.html> 3. <https://docs.aws.amazon.com/redshift/latest/mgmt/copy-unload-iam-role.html> 4. <https://docs.aws.amazon.com/redshift/latest/mgmt/enhanced-vpc-enabling-cluster.html> 5. <https://docs.aws.amazon.com/redshift/latest/gsg/rs-gsg-prereq.html> |
| AWS Network | * A special use case exists for RedShift network isolation and must be noted but requires no action. When database encryption is enabled with KMS, KMS exports a CEK/DEK that is stored on a separate network from the cluster. This network is part of the managed service of RedShift and is not customer configurable or monitored. * Another note to mention is that Redshift clusters exist in another AWS account managed by AWS. This is important to be aware of for monitoring so it is clear what account traffic is going towards and coming from is actually authorized vs rogue. | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-db-encryption.html> |
| **IAM** |  |  |
| Admin Accounts | * A superuser account must be created to perform password disable/enable function (the default user created when a cluster is launched is called **masteruser**) * The IAM entity that creates the cluster is the default owner and first superuser. * A database superuser bypasses all permission checks. Be very careful when using a superuser role. It is recommended that you do most of your work as a role that is not a superuser. Superusers retain all privileges regardless of GRANT and REVOKE commands. * When you launch a new cluster using the AWS Management Console, AWS CLI, or Amazon Redshift API, you must supply a clear text password for the master database user.   To protect the password make sure to encrypt the plaintext password using the following command (assuming CMK used to encrypt cluster) “aws kms encrypt --key-id <kms\_key\_id> --plaintext <password>” | 1. <https://docs.aws.amazon.com/redshift/latest/dg/r_Privileges.html> 2. <https://docs.aws.amazon.com/redshift/latest/dg/r_CREATE_USER.html> |
| Role Based Access Control | * RedShift uses IAM principles to assign rights to actions. When different roles are created and mapped from customer domain groups to AWS IAM roles consider some best practices:   Limit potential over privilege by using redshift:RequestTag Condition key to limit any action to a specific deployment or environment:  {  "Version": "2012-10-17",  "Statement": {  "Sid":"AllowCreateProductionCluster",  "Effect": "Allow",  "Action": "redshift:CreateCluster",  "Resource": "\*"  "Condition":{"StringEquals":{"redshift:RequestTag/usage":"production"}  }  }   * It should go without saying but policies should not include “\*” without having a following deny statement and/or condition statements   For example:  A condition statement to restrict access by redshift:ResourceTag Condition key  {  "Version": "2012-10-17",  "Statement": {  "Sid":"AllowModifyTestCluster",  "Effect": "Allow",  "Action": "redshift:ModifyCluster",  "Resource": "arn:aws:redshift:us-west-2:123456789012:cluster:\*"  "Condition":{"StringEquals":{"redshift:ResourceTag/environment":"test"}  }  }  For example: A deny policy to limit actions to a specific Redshift Cluster environment “production\*”.  {  "Version": "2012-10-17",  "Statement": [  {  "Sid":"AllowClusterManagement",  "Action": [  "redshift:CreateCluster",  "redshift:DeleteCluster",  "redshift:ModifyCluster",  "redshift:RebootCluster"  ],  "Resource": [  "\*"  ],  "Effect": "Allow"  },  {  "Sid":"DenyDeleteModifyProtected",  "Action": [  "redshift:DeleteCluster",  "redshift:ModifyCluster"  ],  "Resource": [  "arn:aws:redshift:us-west-2:123456789012:cluster:production\*"  ],  "Effect": "Deny"  }  ]  }   * Ensure you separate roles that perform snapshot/restore from regular users. Remember to test this as IAM assumes deny but if a broad access role is assigned and a regular user can assume that role then they user can perform the API action. The only way to prevent this is to have a condition statement limiting actions to a specific role or an explicit deny. * Become very familiar with all API actions for RedShift [6] * To properly manage a Service-Linked role be sure to separate the "create" and "delete" actions to unique IAM entity so access to manage data and manage the cluster are separated.   For example: Allow IAM identity to delete Service-Linked Role  {  "Effect": "Allow",  "Action": [  "iam:DeleteServiceLinkedRole",  "iam:GetServiceLinkedRoleDeletionStatus"  ],  "Resource": "arn:aws:iam::<AWS-account-ID>:role/aws-service-role/redshift.amazonaws.com/AWSServiceRoleForRedshift",  "Condition": {"StringLike": {"iam:AWSServiceName": "redshift.amazonaws.com"}}  } | 1. [Managing federation in AWS IAM](https://aws.amazon.com/iam/details/manage-federation/) 2. <https://docs.aws.amazon.com/redshift/latest/mgmt/iam-redshift-user-mgmt.html> 3. <https://docs.aws.amazon.com/redshift/latest/mgmt/redshift-iam-access-control-identity-based.html> 4. <https://docs.aws.amazon.com/redshift/latest/mgmt/redshift-policy-resources.resource-permissions.html> 5. <https://docs.aws.amazon.com/redshift/latest/mgmt/using-service-linked-roles.html> 6. <https://docs.aws.amazon.com/IAM/latest/UserGuide/list_amazonredshift.html> |
| Authorization between AWS services | * Redshift will need to access other AWS services (S3, Glue, Athena, KMS, etc..., to perform functions in an automated way. To setup a role for a specific service function use reference [1]. * It is recommended to run a Service Linked Role for RedShift to limit service specific actions to only the RDS service endpoint [2] * You don't need to manually create an AWSServiceRoleForRedshift service-linked role. Amazon Redshift creates the service-linked role for you. If the AWSServiceRoleForRedshift service-linked role has been deleted from your account, Amazon Redshift creates the role when you launch a new Amazon Redshift cluster.   . | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/authorizing-redshift-service.html> 2. <https://docs.aws.amazon.com/redshift/latest/mgmt/using-service-linked-roles.html> |
| Authentication to AWS platform | RedShift supports local and IAM based authentication. To align to best practice Redshift local users should have passwords disabled which forces authentication based on IAM.  To generate credentials for an IAM user Granting permission to GetClusterCredentials API action should be limited to authorized IAM entities and limited to only specific cluster, database, usernames, and group names.  aws redshift get-cluster-credentials --cluster-identifier examplecluster --db-user temp\_creds\_user --db-name exampledb --duration-seconds 3600   * Db credentials should make use of IAM * This means db users password must be disabled which forces login credentials based on temp IAM credentials | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/options-for-providing-iam-credentials.html> 2. <https://docs.aws.amazon.com/redshift/latest/mgmt/generating-user-credentials.html> |
| Authorization (AWS IAM) of corporate users via Active Directory for access to RedShift resources. | Amazon Redshift requires IAM credentials that AWS can use to authenticate your requests. Those credentials must have permissions to access AWS resources, such as an Amazon Redshift cluster.  AWS Identity and Access Management (IAM) enable organizations with multiple employees to create and manage multiple users, groups and roles under a single AWS account. With IAM policies, companies can grant IAM users/groups/roles fine-grained control to their Amazon RedShift data while also retaining full control over everything the users do.  Most customers have setup federation with AWS accounts. Therefore, the only decision to make is what API actions are needed for roles, groups, or users within IAM [2].  To combine these concepts to control access to RedShift resources, a user would:   * Determine how authentication will occur with Redshift[2] * Create the appropriate IAM roles [5]   For example, a policy to allow IAM users to request credentials for temporary access. The following policy enables the GetCredentials, CreateCluserUser, and JoinGroup actions. The policy uses condition keys to allow the GetClusterCredentials and CreateClusterUser actions only when the AWS user ID matches "AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com". IAM access is requested for the "testdb" database only. The policy also allows users to join a group named "common\_group".  {  "Version": "2012-10-17",  "Statement": [  {  "Sid": "GetClusterCredsStatement",  "Effect": "Allow",  "Action": [  "redshift:GetClusterCredentials"  ],  "Resource": [  "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/${redshift:DbUser}",  "arn:aws:redshift:us-west-2:123456789012:dbname:examplecluster/testdb",  "arn:aws:redshift:us-west-2:123456789012:dbgroup:examplecluster/common\_group"  ],  "Condition": {  "StringEquals": {  "aws:userid":"AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com"  }  }  },  {  "Sid": "CreateClusterUserStatement",  "Effect": "Allow",  "Action": [  "redshift:CreateClusterUser"  ],  "Resource": [  "arn:aws:redshift:us-west-2:123456789012:dbuser:examplecluster/${redshift:DbUser}"  ],  "Condition": {  "StringEquals": {  "aws:userid":"AIDIODR4TAW7CSEXAMPLE:${redshift:DbUser}@yourdomain.com"  }  }  },  {  "Sid": "RedshiftJoinGroupStatement",  "Effect": "Allow",  "Action": [  "redshift:JoinGroup"  ],  "Resource": [  "arn:aws:redshift:us-west-2:123456789012:dbgroup:examplecluster/common\_group"  ]  }  ]  }      }  }   * Map the appropriate AD groups to those roles * Determine if IAM users are allowed to create user credentials within RedShift [4] | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/options-for-providing-iam-credentials.html> 2. [Overview of Managing Access Permissions to Your Amazon Redshift Resources](https://docs.aws.amazon.com/redshift/latest/mgmt/redshift-iam-access-control-overview.html) 3. [Using Identity-Based Policies (IAM Policies) for Amazon Redshift](https://docs.aws.amazon.com/redshift/latest/mgmt/redshift-iam-access-control-identity-based.html) 4. <https://docs.aws.amazon.com/redshift/latest/mgmt/generating-user-credentials.html> 5. <https://docs.aws.amazon.com/redshift/latest/mgmt/redshift-iam-access-control-identity-based.html> |
| Tagging | Amazon Redshift supports tagging to provide metadata about resources at a glance, IAM enforcement, and to categorize your billing reports based on cost allocation.  Best Practice:  Since tags can be used to enforce IAM entity abilities it is important to set appropriate controls on changes to tags with API actions like (DeleteTags, CreateTags)  **Note: tags are not retained if you copy a snapshot to another region, so you must recreate the tags in the new region** | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/amazon-redshift-tagging.html> |
| **Logging & Monitoring** |  |  |
| Logging activity within RedShift | Audit logging is not enabled by default in Amazon Redshift. When you enable logging on your cluster, Amazon Redshift creates and uploads logs to Amazon S3 that capture data from the creation of the cluster to the present time.  To meet logging requirements make sure to enable audit logging:   * Connection log * User log * User activity log (requires additional step after enable of audit logging) * To enable this you must enable the **“enable\_user\_activity\_logging”** database parameter[2]   For example:  aws redshift modify-cluster-parameter-group  --parameter-group-name myclusterparametergroup  --parameters ParameterName=statement\_timeout,ParameterValue=20000 ParameterName=enable\_user\_activity\_logging,ParameterValue=true  As a best practice, after a configuration of RedShift is found to be functional and meet requirements make sure to commit all settings into a [parameter group](https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-parameter-groups.html) so all databases within a cluster are configured the same and each new cluster can be configured the same. (a final deployed cluster should not have parameter group = default.redshift-1.0 because this will not enable logging or other settings specific to customer requirements.) | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/db-auditing.html> 2. <https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-parameter-groups.html> |
| Logging API actions | Cloudtrail will be enabled in every account as part of a default account build. | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/db-auditing.html#rs-db-auditing-cloud-trail> |
| Alerting and Incident Management | You can use the following automated monitoring tools to watch RedShift and report when something is wrong:   * **Amazon CloudWatch Alarms** – Watch a single metric over a time period that you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Auto Scaling policy. CloudWatch alarms do not invoke actions simply because they are in a particular state; the state must have changed and been maintained for a specified number of periods. For more information, see <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/rs-metricscollected.html>. * **AWS CloudTrail Log Monitoring** – Share log files between accounts, monitor CloudTrail log files in real time by sending them to CloudWatch Logs, write log processing applications in Java, and validate that your log files have not changed after delivery by CloudTrail. For more information, see [Logging Amazon RedShift API Calls By Using AWS CloudTrail](https://docs.aws.amazon.com/redshift/latest/mgmt/db-auditing.html#rs-db-auditing-cloud-trail).   **Implementation Note:** To access logs and data for monitoring the data must be decrypted. To decrypt logs/data a customer managed CMK must be defined. Use the same CMK created to encrypt the cluster and create a new policy to grant access only to API actions necessary for tables and actions that are authorized. For example: Use guides in reference [1] where cloudformation templates already exist and can be used to provide a prescriptive approach to collecting and monitoring logs.  Make note of the minimum requirements for access to the Redshift user that is required. Be cautious not to enable more than the necessary “**grant select on all tables in schema pg\_catalog to tamreporting**” entitlement.  **Note**  Audit logging to Amazon S3 is an optional, manual process. When you enable logging on your cluster, you are enabling logging to Amazon S3 only. Logging to system tables is not optional and happens automatically for the cluster. For more information about logging to system tables, see [System Tables Reference](http://docs.aws.amazon.com/redshift/latest/dg/cm_chap_system-tables.html) in the Amazon Redshift Database Developer Guide. | 1. <https://github.com/awslabs/amazon-redshift-monitoring> 2. <https://docs.aws.amazon.com/redshift/latest/mgmt/metrics-listing.html> 3. <https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-event-notifications.html> |
| **Patch/Updates** |  |  |
| Update/Patch for RedShift | When Redshift is configured a maintenance window must be defined to allow for updates to be installed. This window should align with expected change management times and a complete understanding of outages, if any, will occur during this window.  Additional considerations include:   * Refer to <https://docs.aws.amazon.com/cli/latest/index.html> to understand what API action will require a reboot of the cluster or will not. * Understand automatic version updates may change expected behavior and a setting is offered to prevent cluster version changes without approval. | 1. <https://aws.amazon.com/premiumsupport/knowledge-center/notification-maintenance-rds-redshift/> |
| **Availability** |  |  |
| Backup and Restore | * Amazon RedShift makes use of Snapshots to provide customers with a way of recovering to an RPO. Snapshots are stored in Amazon S3, managed by AWS; Snapshots are transferred to S3 over SSL, and where the data in the database is already encrypted in the cluster, it remains encrypted in the snapshot too. [1] * If you enable copying of snapshots from an encrypted cluster and use AWS KMS for your master key, you cannot rename your cluster because the cluster name is part of the encryption context. If you must rename your cluster, you can disable copying of snapshots in the source region, rename the cluster, and then configure and enable copying of snapshots again. [2] * Important Note: If you rotate a DEK/CEK that is used to encrypt a cluster all data will be encrypted with the new key except for snapshots stored in S3. A process should be developed to ensure snapshots are encrypted with the new key to ensure recovery point objectives (RPO) are met. | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-snapshots.html> 2. <https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-db-encryption.html> |
| Limits | Understanding the limitations of the service can help prevent unintentional outages or ability to meet requirements.  Items like:   * # of tables by ec2 instance type * Spectrum limits * Quotas * IAM roles allowed * Naming constraints | 1. <https://docs.aws.amazon.com/redshift/latest/mgmt/amazon-redshift-limits.html> |