Well-Architected Pillars – Guidance Template

Guidance Name: Redshift DevOps

Solution Domain name: Analytics

Guidance Owner/Team: jeesri@/WWSO Analytics team

For each pillar you must answer between one and three of the provided questions. Answers should be within the context of the content produced (e.g avoid referencing things that are not shown to the user). The relevant focus areas alongside the pillar’s whitepaper will be shared with customers.

|  |  |  |  |
| --- | --- | --- | --- |
| Pillar | Question | Answer | Focus Area |
| Cost Optimization | How do you evaluate cost when you select services? | Open source DevOps guidance for Redshift on AWS uses an ec2 instance, Redshift Data warehouse and Git. Github captures and tracks code changes,ec2 instance is used for deploying containers – Jenkins and building image for changed code. Containers allows for packaging Redshift changes and makes them deployable across environments (Dev, Tst and Prod). Redshift and ec2 instance are based on on-demand pricing model, with ability to pause and resume based on guidance needs.  AWS native service guidance for DevOps on Redshift uses code pipeline, ECR and EKS for code deployment. Code pipeline captures code changes and builds the container image, ECR stores the built image for deployment and EKS deploys the container image on a Kubernetes cluster. The deployed container runs like a service to deploy any code changes from code commit. The AWS native service guidance uses serverless service running that need to run when required. | [Cost Effective Resources](https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/cost-effective-resources.html) |
|  | How do you plan for data transfer charges? |  | [Cost Effective Resources](https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/cost-effective-resources.html) |
|  | How do you use pricing models to reduce cost? |  | [Cost Effective Resources](https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/cost-effective-resources.html) |
|  | How does this Guidance scales to continually match the demand and ensure that only the minimum resources required? |  | [Manage Demand and Supply Resources](https://docs.aws.amazon.com/wellarchitected/latest/cost-optimization-pillar/manage-demand-and-supply-resources.html) |
| Security | What design decision have been factored for secure Authentication and Authorisation mechanism, for people and machine access in this Guidance? | The guidance uses IAM service with explicit grants to user and services (ec2, Redshift for open source and code pipeline, ECS and EKS). A .pem file is used to sign on to the ec2 instance, passwords are stored and managed using the AWS secret manager allowing credentials to be audited and rotated automatically. Permissions to services have been granted on a least privilege access model to reduce risk surface. | [Identity and Access Management](https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/identity-and-access-management.html) |
|  | How do you protect resources in this Guidance? |  | [Infrastructure Protection](https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/infrastructure-protection.html) |
|  | How do you protect data in this Guidance? | Data (configuration) has been classified as public (based on sensitivity). Data encryption AES-256 is used to protect data at rest and data in flight making it unintelligible to unauthorized access. | [Data Protection](https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/data-protection.html) |
| Reliability | How does this Guidance implement a highly available network topology? |  | [Foundation](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/foundations.html) |
|  | Are there any limits or constraints that may affect reliability, that the implementer needs to be aware of? |  | [Foundation](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/foundations.html) |
|  | How does this Guidance implement a reliable application-level architecture? (for example: loosely coupled dependencies, throttling, retry limits, stateless compute) | DevOps for Redshift uses purpose built AWS services – ec2, Redshift, code pipeline, EKS and ECS to create a loosely coupled yet tightly integrated architecture. The state of application is stored in code repository (Github, code pipeline) to provide restart ability and stateless compute. | [Architecture](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/workload-architecture.html) |
|  | How does this Guidance implement logs and metrics and send notifications when thresholds are crossed or significant events occur? | Logs and metrics for services are captured in cloud watch log groups. The events are persisted for performance monitoring and troubleshooting. | [Change Management](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/change-management.html) |
|  | How does this Guidance adapt to changes imposed on it, such as changes in demand? |  | [Change Management](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/change-management.html) |
|  | How does this Guidance implement required changes such as deployments and configuration changes? | DDL (Data Definition Language) ,DML (Data manipulation language) and configuration changes are captured using AWS code commit and Github. Each change is captured as a separate branch thereby providing ability to rollbackwards and forwards based on application version requirements. | [Change Management](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/change-management.html) |
|  | How does this Guidance implement data backup and recovery? | Guidance uses s3 to store configurations providing 99.999% availability. EC2 instance , ECS and EKS are used for building and running container instance, have no configuration stored locally. The application is re-deployable in other regions by copying the s3 configuration and creating a new ec2/EKS cluster. | [Failure Management](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/failure-management.html) |
|  | How does this Guidance implement resilience to failures? | S3 is used to maintain resiliency. DevOps guidance maintains a pointer to the last executed DDL/DML statement, this checkpointing in s3 allows the application to restart from the last recovery point. | [Failure Management](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/failure-management.html) |
|  | Does this Guidance enable testing of reliability? |  | [Failure Management](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/failure-management.html) |
|  | Does this Guidance enable recovery from disaster events? |  | [Failure Management](https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/failure-management.html) |
| Operational Excellence | How do you instrument the Guidance to understand its state and achievement of business outcomes? |  | [Prepare](https://docs.aws.amazon.com/wellarchitected/latest/operational-excellence-pillar/prepare.html) |
|  | How did you integrate and deploy changes with the Guidance? | AWS code commit, code build and code deploy are used to capture changes in the repository , build the container image and deploy it respectively. | [Prepare](https://docs.aws.amazon.com/wellarchitected/latest/operational-excellence-pillar/prepare.html) |
|  | How do you safely operate the Guidance and respond to incidents and events? |  | [Operate](https://docs.aws.amazon.com/wellarchitected/latest/operational-excellence-pillar/operate.html) |
|  | How do you implement feedback loops within the Guidance? | Guidance page will be used to collect internal and external feedback. The collected feedback will be used for planning and creating subsequent versions of the guidance. | [Evolve](https://docs.aws.amazon.com/wellarchitected/latest/operational-excellence-pillar/evolve.html) |
| Performance | Why did you select those services, are they purpose built for your use-case? | Open source DevOps guidance for Redshift on AWS uses an ec2 instance, Redshift Data warehouse and Git. Github captures and tracks code changes,ec2 instance is used for deploying containers – Jenkins and building image for changed code. Containers allows for packaging Redshift changes and makes them deployable across environments (Dev, Tst and Prod). Redshift and ec2 instance are based on on-demand pricing model, with ability to pause and resume based on guidance needs.  AWS native service guidance for DevOps on Redshift uses code pipeline, ECR and EKS for code deployment. Code pipeline captures code changes and builds the container image, ECR stores the built image for deployment and EKS deploys the container image on a Kubernetes cluster. The deployed container runs like a service to deploy any code changes from code commit. The AWS native service guidance uses serverless service running that need to run when required. | [Selection](https://docs.aws.amazon.com/wellarchitected/latest/performance-efficiency-pillar/selection.html) |
|  | How can the user experiment with this Guidance and optimize it based on their data? | Guidance used AWS code commit and Github to capture DDL/DML changes. Jenkins container and code build can be used to deploy the changes. Based on data needs, users can create different configurations that get committed and deployed | [Review](https://docs.aws.amazon.com/wellarchitected/latest/performance-efficiency-pillar/review.html) |
|  | How can the location of this Guidance be selected to decrease latency and improve performance? |  | [Network Selection](https://docs.aws.amazon.com/wellarchitected/latest/performance-efficiency-pillar/network-architecture-selection.html) |
|  | How can this Guidance meet the workload requirements of scaling, traffic patterns, data access patterns? | AWS EKS is used to deploy container image as a service, the service scales horizontally based on the traffic pattern and deployment demands. | [Performance Architecture](https://docs.aws.amazon.com/wellarchitected/latest/performance-efficiency-pillar/performance-architecture-selection.html) |
| Sustainability | How does this Guidance scales to continually match the load and ensure that only the minimum resources required? | ? | [User Behavior Patterns](https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/user-behavior-patterns.html) |
|  | How does this Guidance implement architecture patterns for maintaining consistent high utilization of deployed resources? | ? | [Software and Architecture Patterns](https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/software-and-architecture-patterns.html) |
|  | How does this Guidance use technologies that support data access and storage patterns? | ? | [Data Patterns](https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/data-patterns.html) |
|  | How does this Guidance minimize the amount of hardware needed to provision? | ? | [Hardware Patterns](https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/hardware-patterns.html) |