MODULE 16

Praneesh Sharma

21052264

**AWS CDK:**

The AWS CDK is an open-source software development framework that models and provisions cloud application resources using familiar programming languages. The AWS CDK uses the familiarity and expressive power of programming languages for modeling applications. It provides users with high-level components that preconfigure cloud resources with proven defaults, so they can build cloud applications without needing to be an expert. The AWS CDK provisions resources in a safe, repeatable manner through AWS CloudFormation. It also lets users compose and share their own custom components that incorporate their organization's requirements, helping them start new projects faster. The AWS CDK reduces the work of defining and configuring network resources compared to using Amazon CloudFront alone.

Benefits:

The AWS CDK accelerates onboarding to Amazon Web Services (AWS) because there are very few new things to learn. With the AWS CDK, you can use existing skills and tools, and apply those to the task of building cloud infrastructure. It also provides high-level components that preconfigure cloud resources with proven defaults, helping users build on AWS without needing to be an expert. The AWS CDK gives users the expressive power of programming languages for defining infrastructure. Familiar features such as objects, loops, and conditions accelerate the development process. Users can also use the AWS CDK with their integrated development environment (IDE) to take advantage of existing productivity tools and testing frameworks.

With the AWS CDK, users can design their own reusable components that meet their organization’s security, compliance, and governance requirements. As with any other software library, users can share components around their organization, letting them rapidly start up new projects with best practices by default. The AWS CDK lets users build a cloud application without leaving their IDE. Users can write their runtime code and define their AWS resources with the same programming language. Managing infrastructure as code provides great benefits and is often a stepping-stone for a successful application of DevOps practices. In this way, instead of relying on manually performed steps, both administrators and developers can automate provisioning of compute, storage, network, and application services required by their applications using configuration files.

Defining your infrastructure as code makes it possible to:

* Keep infrastructure and application in the same repository
* Make infrastructure changes repeatable and predictable across different environments, AWS accounts, and AWS Regions
* Replicate production in a staging environemnt to permit continuous testing
* Replicate production in a performance test environment used only for the time required to run a stress test.
* Release infrastructure changes using the same tolls as code changes so that deployments include infrastructure updates
* Apply software development bast practice to infrastructure management such as code reviews or deploying small changes frequently.

**Constructs:**

Constructs are cloud components that encode configuration detail, boilerplate, and glue logic for using one or multiple AWS services. The AWS CDK provides a library of constructs that cover many AWS services and features, letting users define their applications' infrastructure at a high level. Additionally, constructs are adjustable and composable. Users can quickly change any of the parameters or encode their own custom construct.

The AWS CDK also provides low-level constructs called *CFN Resources*, which directly represent base-level CloudFormation resources and provide a way to define CloudFormation with a programming language. CFN Resources provide complete coverage of CloudFormation resources and are available shortly after a CloudFormation resource is updated or newly available. With the AWS CDK, anyone can customize, share, and reuse constructs within their organization or community, as with any other software library. This lets users build constructs that help them get started faster and incorporate best practices by default. The AWS CDK lets users define their infrastructure with code and provision it through CloudFormation. They get all the benefits of CloudFormation, including repeatable deployment, rapid rollback, and drift detection.

The AWS CDK lets users model application infrastructure using TypeScript, Python, Java (developer preview), and .NET (developer preview). With the AWS CDK, developers can use existing IDE, testing tools, and workflow patterns. By using tools such as autocomplete and in-line documentation, the AWS CDK lets users spend less time switching between service documentation and their code. The AWS CDK lets users reference their runtime code assets in the same project with the same programming language. For example, they can include their AWS Lambda runtime code or Docker container image in an AWS CDK project and, when they deploy their application, the AWS CDK framework automatically uploads and configures the AWS service with their runtime assets. When the AWS CDK deployment is complete, they will have a fully functional application. The AWS CDK command line interface (CLI) lets users interact with their AWS CDK applications and facilitates functionality such as synthesizing a CloudFormation template, showing the differences between the running stack and proposed changes, confirming security-related changes before deployment, and deploying multiple stacks across multiple environments.