[#cdk-pipeline-security] == Security notes

Any form of continuous delivery has inherent security risks. Under the {aws} https://aws.amazon.com/compliance/shared-responsibility-model/[Shared Responsibility Model], you are responsible for the security of your information in the {aws} Cloud. The CDK Pipelines library gives you a head start by incorporating secure defaults and modeling best practices.

However, by its very nature, a library that needs a high level of access to fulfill its intended purpose cannot assure complete security. There are many attack vectors outside of {aws} and your organization.

In particular, keep in mind the following:

* Be mindful of the software you depend on. Vet all third-party software you run in your pipeline, because it can change the infrastructure that gets deployed.
* Use dependency locking to prevent accidental upgrades. CDK Pipelines respects package-lock.json and yarn.lock to make sure that your dependencies are the ones you expect.
* CDK Pipelines runs on resources created in your own account, and the configuration of those resources is controlled by developers submitting code through the pipeline. Therefore, CDK Pipelines by itself cannot protect against malicious developers trying to bypass compliance checks. If your threat model includes developers writing CDK code, you should have external compliance mechanisms in place like https://aws.amazon.com/blogs/mt/proactively-keep-resources-secure-and-compliant-with-aws-cloudformation-hooks/[{aws} CloudFormation Hooks] (preventive) or https://aws.amazon.com/config/[{aws} Config] (reactive) that the {aws} CloudFormation Execution Role does not have permissions to disable.
* Credentials for production environments should be short-lived. After bootstrapping and initial provisioning, there is no need for developers to have account credentials at all. Changes can be deployed through the pipeline. Reduce the possibility of credentials leaking by not needing them in the first place.