Environment Configurations for Code Engine

Configuring the environment for your Code Engine projects is crucial for customization and security. Environmental variables play a key role, allowing you to tailor your project's environment and safeguard sensitive information like passwords and tokens.

Best practices for handling secrets in code engine

Avoid placing passwords, tokens, or confidential details in your code, as anyone with read access to your project can easily see this information. A more secure approach is to store such secrets in your project's environment variables, where only project collaborators and administrators can access them. The access level may vary depending on the scope defined earlier for the variable.

Environment variables (env variables) can be configured as key-value pairs, usable by your application, job, or function. These variables can be defined as literal values or as references to existing secrets or configmaps. Code Engine utilizes environment variables to facilitate the use of secrets or configmaps by your job, application, or function.

Automatically injected variables for jobs

- When running a job, Code Engine automatically injects essential environment variables into the job run instance.
- Each job run instance gets a unique index from the specified array of indices, starting from 0, assigned by Code Engine. The **JOB_INDEX** environment variable contains this index.
- While jobs themselves don't have URLs, CE_DOMAIN and CE_SUBDOMAIN values become handy for referencing applications that are running in the same project.
- The full external URL of this application is **appName.CE_SUBDOMAIN.CE_DOMAIN**. To reference the private URL of an application, use **appName.CE_SUBDOMAIN**.

The list of automatically injected environment variables can be found here.

Working with environmental variables

Managing environment variables in Code Engine is a breeze, and you can do it in two ways:

1. Console method:
    - Define environment variables while creating your app, job, or function through the Code Engine console.
    - Decide if you want to create a literal environment variable or one that references an existing secret or configmap. If you want your environment variable to fully reference an existing secret or configmap or reference individual keys in an existing secret or configmap, the secret or configmap must exist.
    - Use the console for creating, updating, or deleting environment variables. Step-by-step instructions for creating, updating/modifying, or deleting environment variables can be found here for your guidance.
2. CLI method:
    - Utilize the Code Engine CLI to create and manage environment variables.
    - Define variables when creating or updating your app, job, or function using the CLI.
    - Whether referencing configmaps or secrets or creating a literal variable, the CLI offers flexibility.
    - Before you start, ensure your Code Engine CLI environment is set up.
    - The environment variables can be created, updated, or deleted using the CLI. For detailed instructions for creating, updating/modifying, or deleting when you no longer need it, see here and follow the step-by-step guide.

Conclusion

Understanding and utilizing environment configurations in Code Engine is important for tailoring your project's environment and safeguarding sensitive information. Whether you prefer the user-friendly console or the command line, the process is simplified for creating, updating, or deleting variables. Mastering these configurations enhances your ability to create dynamic and secure applications effortlessly.