HCLTech | Supercharging Progress™

AWS Ecosystem Business Unit – Generative AI COE

Code GReat (Generate, Review and Execute Code)

Overview

The use of Generative AI to assist in code development and code deployment is garnering huge interest across organizations. Integrated development environment (IDE) tools like Amazon Code Whisperer and Microsoft GitHub Copilot that leverage large language models are being evaluated and utilized to accelerate development work. Organizations are looking at solutions that provide efficiency gains in the form of increased developer productivity and this is where Code GReat comes to the rescue.

Code GReat solution utilizes a large language model-enabled graphical user interface to facilitate natural language interactions for code generation, review and execution. Users can generate programming code, review it and execute it in a conversational way. Code GReat aims to streamline the development process by leveraging the power of Generative AI.

Features:

01   Multi-language code generation –Translate natural language prompts into source code across various programming languages like Python, Java, JavaScript etc. as well as infrastructure as code (IAC) templates like Terraform, CloudFormation etc.

02   Code analysis - Perform code reviews and code hygiene checks to identify potential issues related to security, performance, reliability etc.

03   LLM integration - Provides the option to leverage various Large Language Models (LLMs) like ClaudeV2 via integration with AWS services like Bedrock and SageMaker Jumpstart.

04   Flexible code generation - Generate code snippets and templates for many different scenarios like deploying cloud infrastructure, querying databases, finding files in Linux etc.

05   Extensible architecture - Has the potential to be extended further to execute the generated code automatically, enabling full end-to-end infrastructure provisioning and configuration from natural language prompts.

Solution benefits:

Improved developer productivity

Automates time-consuming coding tasks and enables developers to focus on complex application logic.

Better code quality

Identifies issues early through automated code reviews and hygiene checks, leading to more secure and reliable applications.

Reduced costs

Cuts down on developer hours required for repetitive coding tasks like configs and queries.

Code consistency

Promotes code consistency across development teams as developers can iterate and land the right prompt to generate the expected code.

Learning

Accelerate the learning process for cloud programming languages thereby reducing resource ramp-up time

Increased adoption

By bringing together key capabilities for developers, IT operations, quality assurance and admininstrators, this solution helps organizations deliver higher-quality software faster and more efficiently.

- DevOps engineers can create scripts to automate deployment pipelines for faster release cycles.
- Quality assurance specialists can use code analysis features to review code quality and adherence to best practices.
  Systems administrators can deploy monitoring and notification services to manage infrastructure resources effectively.

Technical architecture:
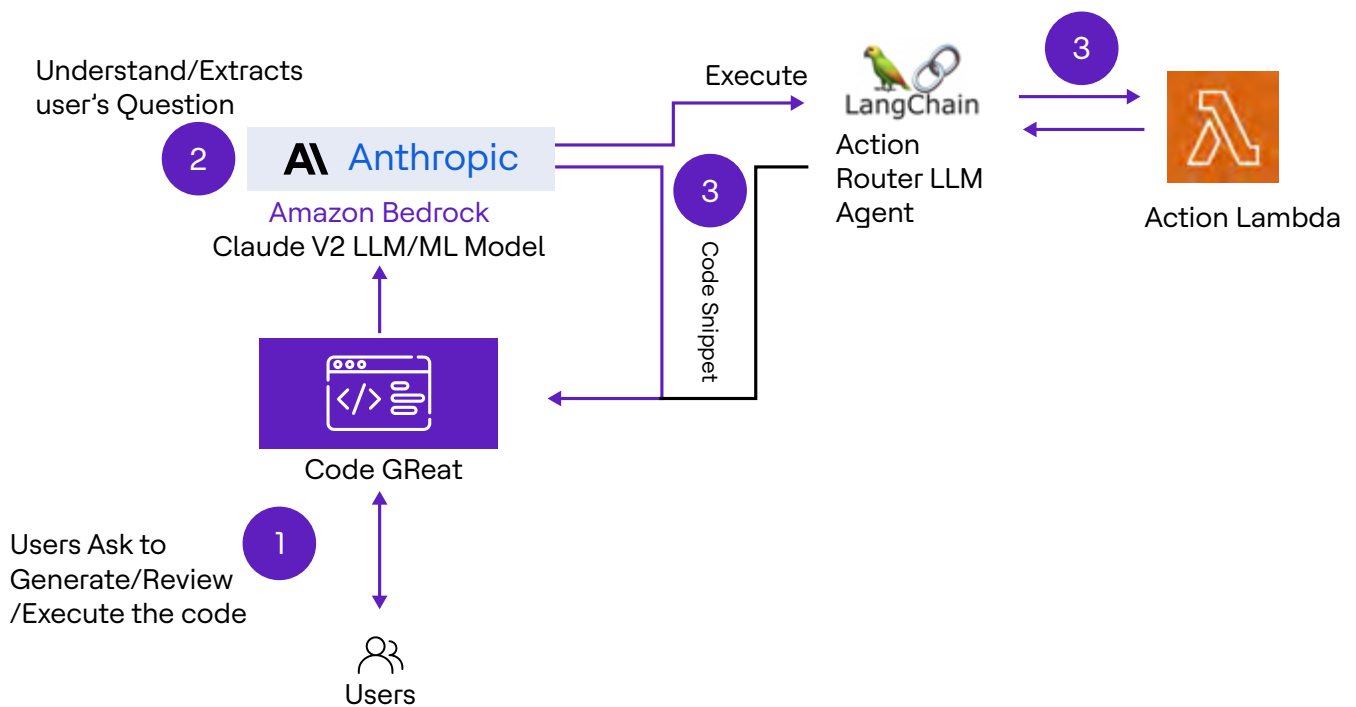
01 The developer or system administrator inputs their query in natural language through the conversational interface. This allows them to phrase their request in plain terms without needing knowledge of specific query syntax.

02 Query parsing- Query parser takes the natural language query, parses it with required prompt settings and sets the context for Amazon Bedrock powered LLM to generate the response.

03 Response- The LLM generates a technical response including code snippets, explanations and relevant references tailored to the query.

04 The solution can be extended to automatically execute generated code like CloudFormation scripts, providing end-to-end automation of tasks through natural language interaction.

HCLTech | Supercharging Progress™

hcltech.com