

How to use private repositories in GO projects



Ayush Yadav · Follow

3 min read · May 8, 2024

 14

 1







Overview:

In this guide, we'll explore how to utilize private repositories in your Go projects. We'll cover the steps needed to configure your Go environment to seamlessly integrate private repositories.

Prerequisites:

- Ensure you have Go installed on your system.
- You should have ssh keys setup linked to your Github account.

Go Projects:

- We have 2 Go projects/ repositories here.
- The Go private repository you want to use.
- The new Go project you want to use the private repository in.
- Eg: Using a private cleaner repo in a new data pipeline Go project.

• • •

Step-by-Step Guide

Step 1. Get the private repository path:

- Begin by navigating to your private repository's *go.mod* file.
- Here you will find the full path to your private repository, which is essential for the import process.
- The path usually follows the format “github.com/user/private_repo”

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It displays Go code for configuring a private repository.

```
/* the go.mod file of private repository you are
trying to use */

module github.com/user/private_repo

go 1.19
```

Step 2: Configure Go for Private Repositories:

To let Go know to use your private repo instead of fetching it from public servers we export GOPRIVATE variable with the full repository path.

```
export GOPRIVATE="github.com/username/private_repo"
```

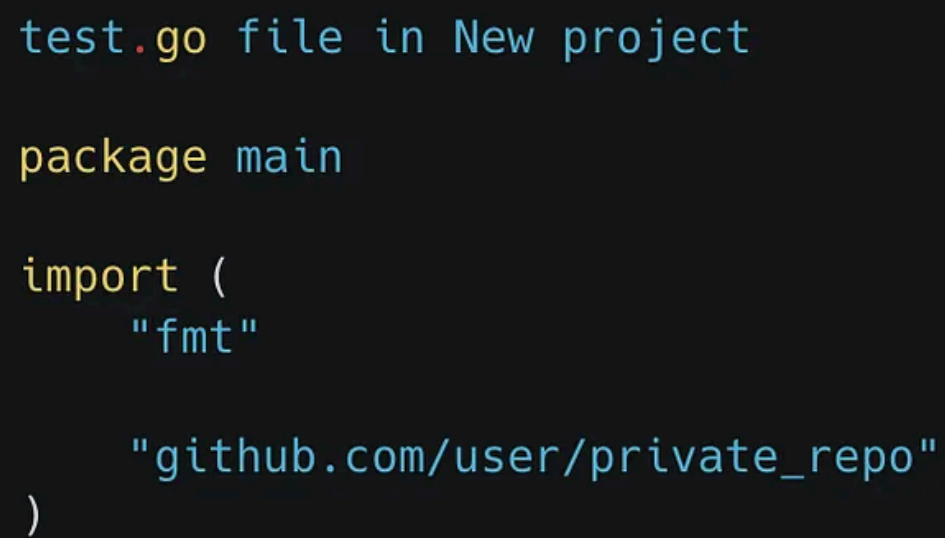
Step 2: Import the private repository in your new project:

Open any Go file in your new project where you need to import your private package. For example, if you're working with a file named test.go, you would include the following import statement at the top:

```
// eg: test.go file in the new project

package main

import (
    "fmt"
    "github.com/username/private_repo"
)
```



```
test.go file in New project

package main

import (
    "fmt"

    "github.com/user/private_repo"
)
```

3. Use SSH for authentication:

Run these command in terminal, to use ssh instead of http while cloning the repos

```
git config --global url."git@github.com:".insteadOf "https://github.com/"
```

Step 4: Tidy Up Your go.mod File:

This command cleans up your project's go.mod file, adding the private repository you're importing, updating the go.sum file, and downloading the necessary dependencies to your system.



Step 5: Run Your Go File

You can access any function in the new project by using the package name of the imported repo followed by function name.

eg: `package_name.DoSomething()`

```
// File: test.go

package main

import (
    "github.com/user/private_repo"
)

func main() {
    private_repo.DoSomething()
}

// run file using -> go run test.go
```

```
# run new project using  
go run test.go  
  
# in the test.go file use  
functions using  
  
package_name.DoSomething( )
```

Conclusion:

By following these steps, we've successfully imported a private Go repository into our project.

For more insights on Generative AI, Python, Systems, and indie product development, follow my journey.

Happy coding, and may your Go projects run smoothly and securely!!

[Go](#)[Golang](#)[Private Repository](#)



Written by Ayush Yadav

126 Followers · 34 Following

Follow

Generative AI Wizard | Building the future one pipeline at a time.

Responses (1)



What are your thoughts?

Respond



SHIVAM SOURAV JHA
Sep 5, 2024

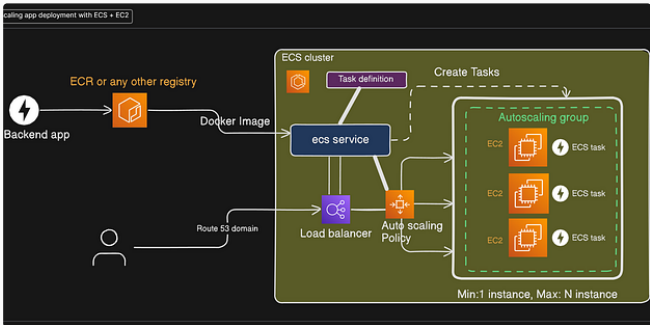


fun fact I was about to write on this because I faced this. Searched for it and found out that you're written about it



Reply

More from Ayush Yadav

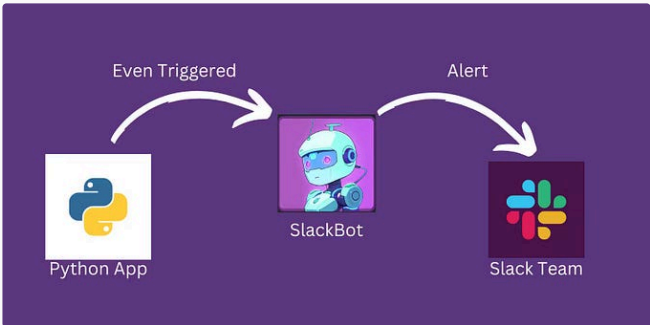


Ayush Yadav

How to Build ECS + EC2 Auto-Scaling Infrastructure on AWS

Learn to deploy a scalable backend using ECS and EC2 on AWS. This guide covers...

Jun 15, 2024 126 5

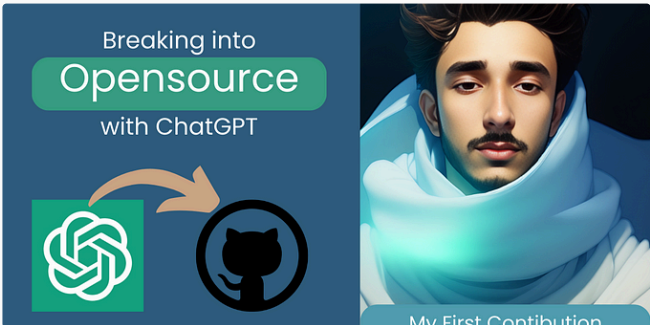
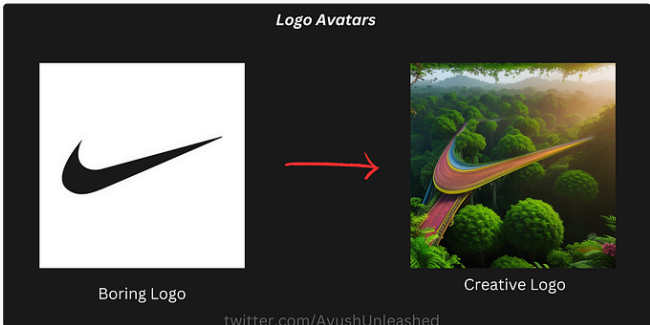



Ayush Yadav

A Beginner's Guide to Building Slack Bots

Python | Slack bot for alerts

Jan 7, 2024 53



 Ayush Yadav

How to Build Your First GenAI App with Stable Diffusion and Control...


A Step-by-Step Guide

Jan 7, 2024

 432

 3



 Ayush Yadav

Breaking into Open-Source with ChatGPT’s Help: My First...

Open-source projects are a great way to get involved in the software development...

Jan 7, 2024

 12



See all from Ayush Yadav

Recommended from Medium



In The Ordinary Programmer by huizhou92

Error Handling in Go: The New ? operator

It’s Not A Silver Bullet.



Jan 11

 123

 16



Feature	Unbuffered Channel	Buffered Channel
Capacity	0 (no buffer)	> 0 (specified capacity)
Locking behavior	Sender/receiver block until the other side is ready	Sender blocks only if the buffer is full, and receiver blocks only if the buffer is empty
Communication type	Synchronous	Asynchronous
Use Case	Synchronization between goroutines	Decoupling of sender and receiver



In Breaking Tech by SHIVAM SOURAV JHA

Go Channels: Surprised me yet again

In tech there is one technical description and one practical explanation. This weekend I...



Jan 5

 114



Lists



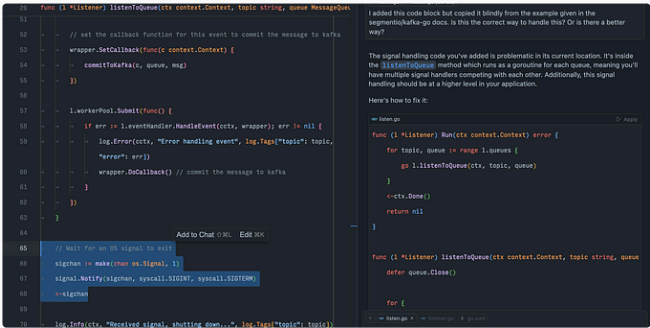
General Coding Knowledge

20 stories · 1879 saves



Yash

Go 1.24: The Ultimate Update That Makes Development Faster and...



In Level Up Coding by Jacob Bennett

The 5 paid subscriptions I actually use in 2025 as a Staff Software...

Go 1.24 (scheduled for release in February 2025) brings a variety of powerful new...

★ Jan 12 🖱️ 196 📌⁺



🌟 In loveholidays tech by George Smith

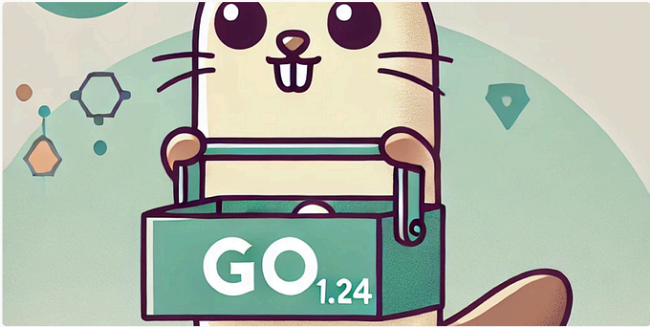
go-config-loader: A Simple Approach to Go Configuration...

Configuration management is a common challenge in software development,...

Jan 8 🖱️ 284 📌⁺

Tools I use that are cheaper than Netflix

★ Jan 7 🖱️ 5.7K 💬 121 📌⁺



🌐 Stefanie Lai

Exploring Go 1.24: What's New and Why It Matters

A Preview of Go 1.24 Innovations

★ Jan 7 🖱️ 93 💬 3 📌⁺

See more recommendations