VStack, HStack, and other container views in SwiftUI are all implemented using the `@ViewBuilder` attribute. This means that you can use them to combine multiple views into a single view, and you can also use them to create dynamic views.

For example, the following code creates a VStack that contains two Text views:

```Swift
VStack {
    Text("Hello, world!")
    Text("This is a second text view.")
}
```

The VStack view in this code is a @ViewBuilder, which means that it can combine multiple views into a single view. The VStack view will stack the two Text views vertically on top of each other.

You can also use @ViewBuilder to create dynamic views. For example, the following code creates a VStack that contains a Text view with dynamic text:

```Swift
@ViewBuilder
func myView() -> some View {
    let text = "Hello, world!"
    return VStack {
        Text(text)
    }
}
```

The VStack view in this code is a @ViewBuilder, and the text inside the VStack is dynamic. This means that the text will change if the value of the `text` variable changes.

Using @ViewBuilder to implement container views in SwiftUI has a number of benefits. First, it makes it easy to create new types of container views. Second, it makes it easy to create dynamic views. Third, it makes it easy to test views, because you can mock or stub the implementations of the protocol's methods. Fourth, it makes it easier to write reusable code, because you can write functions that accept any type of View that conforms to the View protocol.

Overall, `@ViewBuilder` is a powerful tool that makes it easy to create and use complex views in SwiftUI.