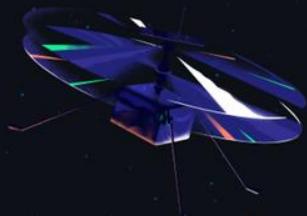




GitHub Copilot for DevOps



DevOps

AGENDA

- Capabilities & Features
- Common Problems
- Hands-on Lab
- Q & A

Outcome of this training

At the end of training, you will achieve the following results...

- ✓ Get answers to specific use case scenario questions
- ✓ Increase existing Copilot skills by following a specific workshop tutorial
- ✓ Learn in-depth tips and tricks and best practices on how to best utilise GitHub Copilot



Workshop

Capabilities and Features

Docker

Quickly generate
a **Dockerfile** using
GitHub Copilot

Dockerfile

```
1  # a dockerfile based on the official node image to build and run the app
2
3  # use the official node image
4  FROM node:latest
5
6  # set the working directory
7  WORKDIR /usr/src/app
8
9  # copy the package.json and package-lock.json files
10 COPY package*.json ./
11
12 # install the dependencies
13 RUN npm install
14
15 # copy the rest of the files
16 COPY . .
17
18 # expose the port
19 EXPOSE 3000
20
21 # run the app
22 CMD ["npm", "start"]
```

Infrastructure as Code

Ask Copilot to
generate generate
IaC templates

```
terraform > 🌐 main.tf > 🏛 resource "azurerm_app_service_plan" "appserviceplan" > 🏛 sku
1   # Azure App Service linux hosted Node.js application in australia east
2
3   resource "azurerm_app_service_plan" "appserviceplan" {
4     name          = "appserviceplan"
5     location      = "australiaeast"
6     resource_group_name = azurerm_resource_group.rg.name
7     kind          = "Linux"
8     reserved      = true
9
10    sku {
11      tier = "Standard"
12      size = "S1"
13    }
14  }
15
16  resource "azurerm_app_service" "appservice" {
17    name          = "appservice"
18    location      = "australiaeast"
19    resource_group_name = azurerm_resource_group.rg.name
20    app_service_plan_id = azurerm_app_service_plan.appserviceplan.id
21
22    site_config {
23      linux_fx_version = "DOCKER|node:12.16.1-alpine"
24    }
25  }
```

CI/CD Pipelines

Copilot can help
to generate your
CI/CD pipelines
too

```
1 # build and deploy an azure function
2 name: Build and Deploy Azure Function
3
4 on:
5   push:
6     branches:
7       - main
8   pull_request:
9     branches:
10      - main
11
12 jobs:
13
14   build:
15     runs-on: ubuntu-latest
16     steps:
17       - uses: actions/checkout@v2
18       - name: Setup .NET Core
19         uses: actions/setup-dotnet@v1
20         with:
21           dotnet-version: 3.1.101
22       - name: Build with dotnet
23         run: dotnet build --configuration Release
24       - name: Test with dotnet
25         run: dotnet test --no-build --verbosity normal
26       - name: Publish with dotnet
27         run: dotnet publish --configuration Release --output ./bin/Release/netcoreapp
28       - name: Deploy to Azure Function App
29         uses: azure/webapps-deploy@v2
30         with:
31           app-name: 'my'
32           slot-name: 'production'
```

Write Scripts

Helps with **script writing** and reading that **cryptic bash syntax**.

The screenshot shows a GitHub Copilot interface with the following details:

- Code Snippet:** `@workspace /explain`
- Copilot Information:** GitHub Copilot used @workspace. A note says "Copilot generated code may be incorrect".
- Line Number:** 3
- Code Line:** `echo "My parent folder is: ${PWD%/*}"`
- Explanation:** The selected code is a single line of a Bash shell script. It uses the `echo` command to print a message to the terminal. The message includes the value of a special variable, `PWD`, which stands for "Present Working Directory". This variable always holds the full path of the current directory.

The `${PWD%/*}` expression is a form of parameter expansion in Bash. The `%` operator trims the shortest match from the end of the string. In this case, it trims everything from the last slash (`/`) to the end of the string. This effectively removes the current directory from the `PWD` path, leaving only the parent directory.

So, the entire line `echo "My parent folder is: ${PWD%/*}"` prints a message that states the path of the parent directory of the current working directory. For example, if the current directory is `/home/user/documents`, the script will print `My parent folder is: /home/user`.

Alerts and Monitoring

Quickly generate
IaC alert and
monitoring
configuration

```
// App Service
> resource "azurerm_app_service" "app_service" { ...
}

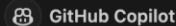
// App Service Plan
> resource "azurerm_app_service_plan" "app_service_plan" { ...
}

// Alert Rule
resource "azurerm_monitor_metric_alert" "alert_rule" {
  name          = "copilot-alert-rule"
  resource_group_name = azurerm_resource_group.rg.name
  scopes        = [azurerm_app_service.app_service.id]
  description   = "Alert rule for copilot app"
  severity      = 3
  window_size   = "PT5M"
  frequency     = "PT1M"
  criteria {
    metric_namespace = "Microsoft.Web/sites"
    metric_name      = "Requests"
    aggregation      = "Total"
    operator         = "GreaterThan"
    threshold        = 10
    dimension {
      name      = "ResultType"
      operator = "Include"
      values   = ["Success"]
    }
  }
  action {
    action_group_id = azurerm_monitor_action_group.action_group.id
  }
}
```

Log Analysis

Help investigate and identify issues in log files

what's the reason for exceptions in this file



Used 1 reference >

From the provided log excerpt, there are two types of exceptions:

1. **Database connection timeout on query**: This error typically occurs when a database query takes longer than the configured timeout duration. This could be due to network issues, database server performance, or the complexity of the query itself.
2. **NullReferenceException in CartController.AddToCart**: This error occurs when you try to access a member on a type whose value is `null`. In this context, it's happening in the `AddToCart` method of the `CartController`. This could be due to an uninitialized object, a missing value that the method expects, or a failed method call that returns `null`.

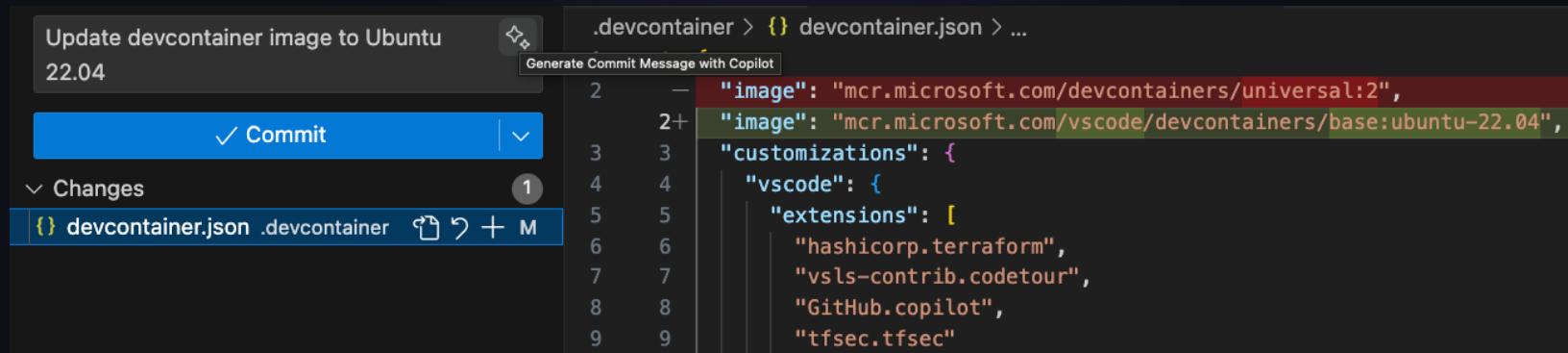
Please note that these are just the error messages. To find the exact cause of these exceptions, you would need to look at the stack trace (if available) or debug the application.

```
1 2024-01-31 09:02:00 DEBUG:: Initializing database connections.
3 2024-01-31 09:03:00 INFO:: HTTP GET Request for '/home/index'.
4 2024-01-31 09:04:00 WARN:: Low memory warning.
5 2024-01-31 09:05:00 ERROR:: Database connection timeout on query.
6 2024-01-31 09:06:00 INFO:: HTTP POST Request for '/account/login' with status 200 OK.
7 2024-01-31 09:07:00 DEBUG:: User authentication successful for user 'admin'.
8 2024-01-31 09:08:00 ERROR:: NullReferenceException in CartController.AddToCart.
9 2024-01-31 09:09:00 INFO:: Scheduled task 'UpdateProductCatalog' started.
10 2024-01-31 09:10:00 INFO:: Scheduled task 'UpdateProductCatalog' completed successful
11 2024-01-31 09:11:00 INFO:: Application start.
12 2024-01-31 09:12:00 DEBUG:: Initializing database connections.
13 2024-01-31 09:13:00 INFO:: HTTP GET Request for '/home/index'.
14 2024-01-31 09:14:00 WARN:: Low memory warning.
15 2024-01-31 09:15:00 ERROR:: Database connection timeout on query.
16 2024-01-31 09:16:00 INFO:: HTTP POST Request for '/account/login' with status 200 OK.
17 2024-01-31 09:17:00 DEBUG:: User authentication successful for user 'admin'.
18 2024-01-31 09:18:00 ERROR:: NullReferenceException in CartController.AddToCart.
19 2024-01-31 09:19:00 INFO:: Scheduled task 'UpdateProductCatalog' started.
20 2024-01-31 09:20:00 INFO:: Scheduled task 'UpdateProductCatalog' completed successful
21 2024-01-31 09:21:00 INFO:: Application start.
22 2024-01-31 09:22:00 DEBUG:: Initializing database connections.
23 2024-01-31 09:23:00 INFO:: HTTP GET Request for '/home/index'.
24 2024-01-31 09:24:00 WARN:: Low memory warning.
25 2024-01-31 09:25:00 ERROR:: Database connection timeout on query.
26 2024-01-31 09:26:00 INFO:: HTTP POST Request for '/account/login' with status 200 OK.
27 2024-01-31 09:27:00 DEBUG:: User authentication successful for user 'admin'.
28 2024-01-31 09:28:00 ERROR:: NullReferenceException in CartController.AddToCart.
```

Helping you become a shell
superhero...

Commit messages

Automatically generate your **commit messages**



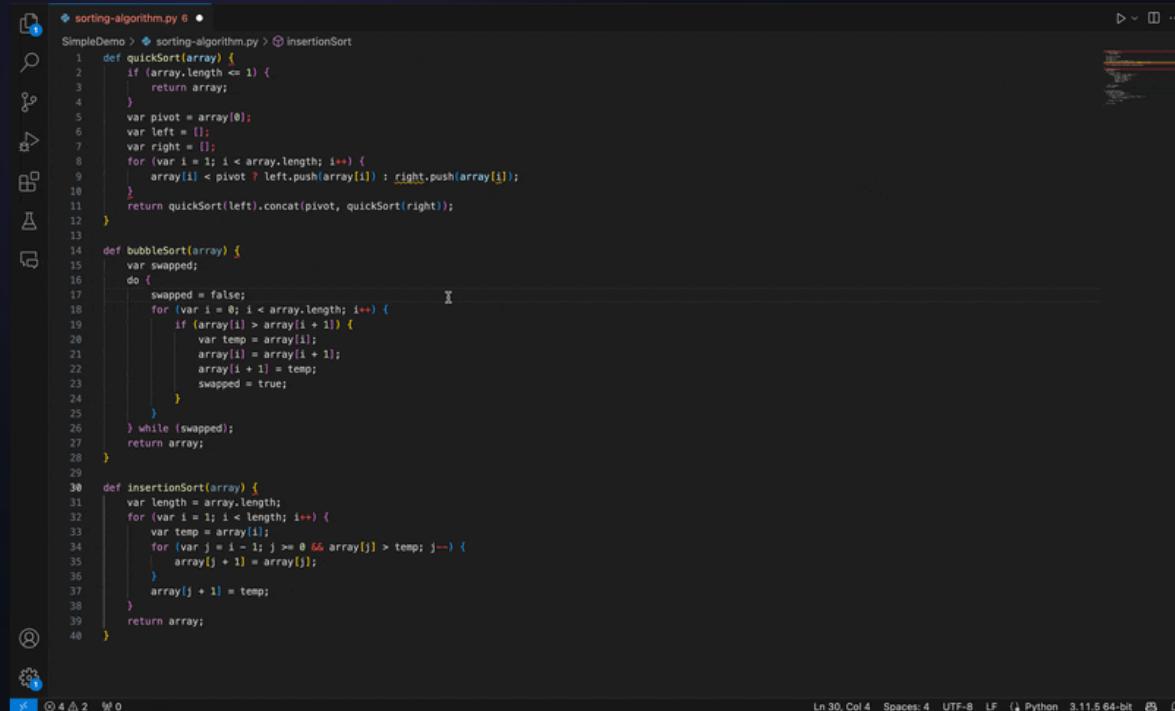
A screenshot of a GitHub commit interface. The commit message is "Update devcontainer image to Ubuntu 22.04". The "Commit" button is highlighted in blue. A Copilot icon is visible above the commit message input field. The commit message content is as follows:

```
Generate Commit Message with Copilot
Update devcontainer image to Ubuntu 22.04
✓ Commit
Changes
{} devcontainer.json .devcontainer
2 - "image": "mcr.microsoft.com/devcontainers/universal:2",
2+ "image": "mcr.microsoft.com/vscode/devcontainers/base:ubuntu-22.04",
3 3 "customizations": {
4 4   "vscode": {
5 5     "extensions": [
6 6       "hashicorp.terraform",
7 7       "vsls-contrib.codetour",
8 8       "GitHub.copilot",
9 9       "tfsec.tfsec"

```

In-file Copilot options

Copilot now offers **in-file Copilot feature** to selectively improve

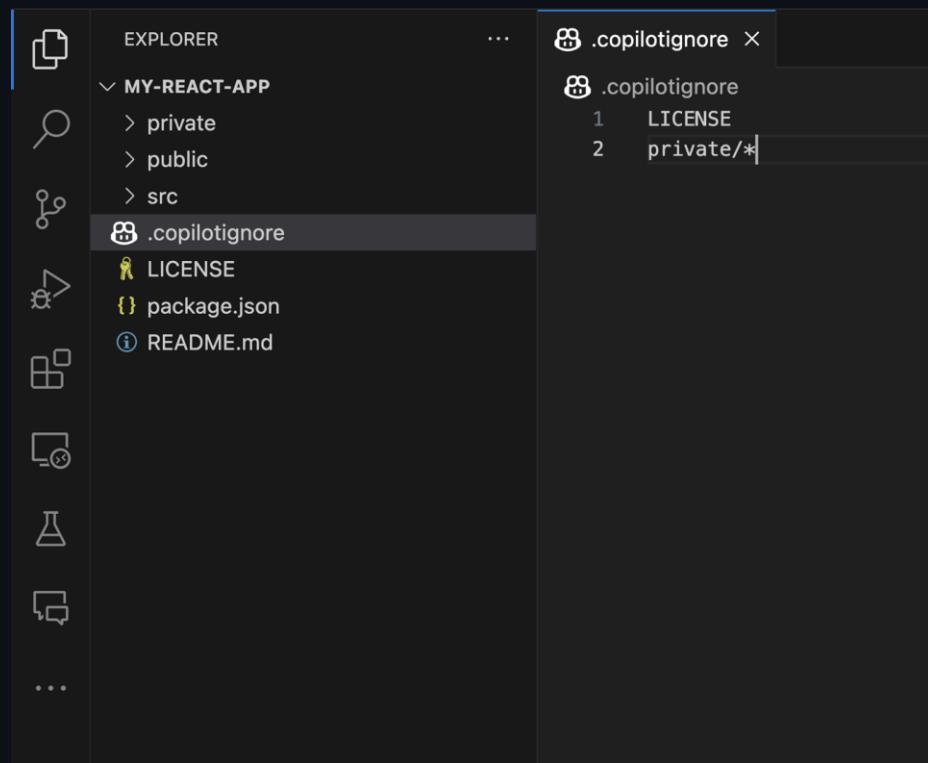


The screenshot shows a code editor window with a dark theme. The file being edited is `sorting-algorithm.py`. The code contains three sorting functions: `quickSort`, `bubbleSort`, and `insertionSort`. The Copilot feature is active, indicated by blue underlines and small blue icons next to certain lines of code, particularly in the `bubbleSort` and `insertionSort` functions. The status bar at the bottom right shows the file path as `SimpleDemo > sorting-algorithm.py > insertionSort`, the line number as `Ln 30, Col 4`, and the character count as `Spaces: 4`.

```
1 def quickSort(array) {
2     if (array.length <= 1) {
3         return array;
4     }
5     var pivot = array[0];
6     var left = [];
7     var right = [];
8     for (var i = 1; i < array.length; i++) {
9         if (array[i] < pivot) {
10             left.push(array[i]);
11         } else {
12             right.push(array[i]);
13         }
14     }
15     return quickSort(left).concat(pivot, quickSort(right));
16 }
17
18 def bubbleSort(array) {
19     var swapped;
20     do {
21         swapped = false;
22         for (var i = 0; i < array.length; i++) {
23             if (array[i] > array[i + 1]) {
24                 var temp = array[i];
25                 array[i] = array[i + 1];
26                 array[i + 1] = temp;
27                 swapped = true;
28             }
29         }
30     } while (!swapped);
31     return array;
32 }
33
34 def insertionSort(array) {
35     var length = array.length;
36     for (var i = 1; i < length; i++) {
37         var temp = array[i];
38         for (var j = i - 1; j >= 0 && array[j] > temp; j--) {
39             array[j + 1] = array[j];
40         }
41         array[i] = temp;
42     }
43     return array;
44 }
```

Block files from Copilot

Use `.copilotignore` to block files and folders from being used by Github Copilot



Block files from Copilot

Ignore specific repositories and paths in Restricted Content

The screenshot shows the GitHub organization settings for 'evil-copilot'. The left sidebar lists various settings categories like General, Features, Access, and Copilot. The 'Copilot' section is expanded, and 'Restricted content' is selected. A 'Beta' button is visible next to 'Restricted content'. The main area displays a code editor with YAML syntax for specifying ignored paths. The code is as follows:

```
1  `/*`*: ["**/.env"]  
2  
3  marais-testing-repo:  
4    - "/src/kernel.rs"  
5  
6  git@github.com:prime/react.git:  
7    - "e2e/*"  
8  
9  git@gitlab.com:openicity-labs/surreal-react.git:  
10   - "fly.toml"  
11   - "**/surrealdbjs/*"
```

Below the code editor, there's a note about keyboard navigation and a save button at the bottom right.

Set the scene for GitHub Copilot

Set the context for suggestions using **#file**, **#select**, or **@workspace**

The screenshot shows the GitHub Copilot interface integrated into a code editor. On the left, a 'CHAT' window displays a user command: '@workspace /explain #file:BankAccount.cs How can I implement authentication in the selected code'. Below this, a message from 'GitHub Copilot' provides context: 'Used 2 references > The selected code already has a basic form of authentication implemented. Here's how it works:'. It lists two points:

1. The `BankAccount` class has a private field `password` that is set when a new `BankAccount` object is created. This is done in the constructor method `BankAccount`.

```
private string password;  
public BankAccount(string accountNum  
{  
    // Existing code...  
    this.password = password;  
}
```
2. The `Authenticate` method checks if the provided password matches the password stored in the `password` field.

```
public bool Authenticate(string passw  
{  
    return this.password == password;  
}
```

On the right, the full C# code for `BankAccount.cs` is shown, including the constructor and the `Authenticate` method.

```
C# BankAccount.cs X  
C: > Temp > C# BankAccount.cs > BankAccount  
1 reference  
1 public class BankAccount  
2 {  
3     // Existing code...  
4  
5     2 references  
6     private string password;  
7  
8     0 references  
9     public BankAccount(string accountNumber,  
10    {  
11        // Existing code...  
12        this.password = password;  
13    }  
14  
15     1 reference  
16     public bool Authenticate(string password)  
17    {  
18        // Existing code...  
19        return this.password == password;  
20    }  
21  
22     0 references  
23     public void Credit(double amount, string  
24    {  
25        if (!Authenticate(password))  
26        {  
27            throw new Exception("Authenticati  
28        }  
29  
30        // Existing code...  
31    }  
32  
33        // Similar changes would be made to the D  
34    }
```

Terminal Commands

@terminal to suggest terminal commands from Copilot chat.

@terminal deploy a node js application to an azurer app service

 GitHub Copilot used @terminal

Used 15 references >

To deploy a Node.js application to Azure App Service, you can use the Azure CLI. Here's a step-by-step command:

1. Login to Azure:
`az login`
2. Create a resource group (if it doesn't exist):
`az group create --name {resource_group_name} -`
3. Create an App Service plan:
`az appservice plan create --name {plan_name} -`
4. Create a web app:
`az webapp create --resource-group {resource_gr`
5. Deploy your Node.js app:

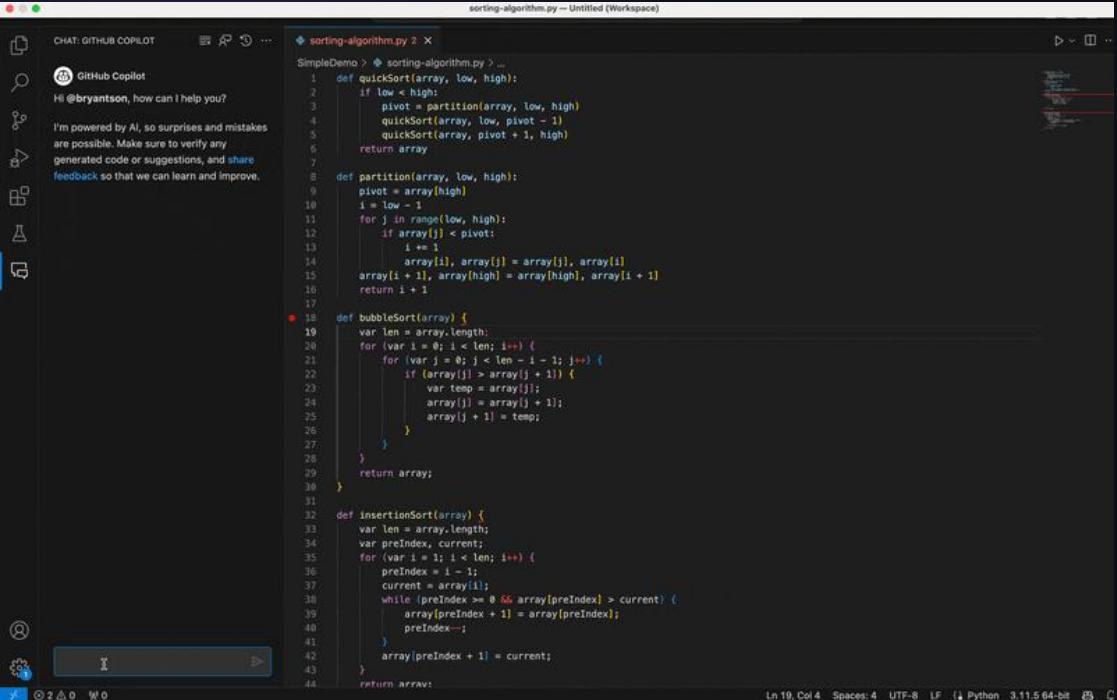
Search

@vscode /search
to generate search
terms and patterns
for Visual Studio
Code

The screenshot shows a dark-themed interface, likely GitHub Copilot, with a search bar at the top containing the text "@vscode /search fine all files containing an email address". Below the search bar, there is a result card for "GitHub Copilot used @vscode". The card includes a small icon of a person with a gear, the text "GitHub Copilot used @vscode", and a detailed explanation: "To find all files containing an email address, you can use the regex [\w.-]+@[\\w.-]+\\.\\w+ . This will match most common email formats." At the bottom of the card is a blue "Search" button.

/fix, /tests and other / command options

Leverage / command options that can help to improve your code



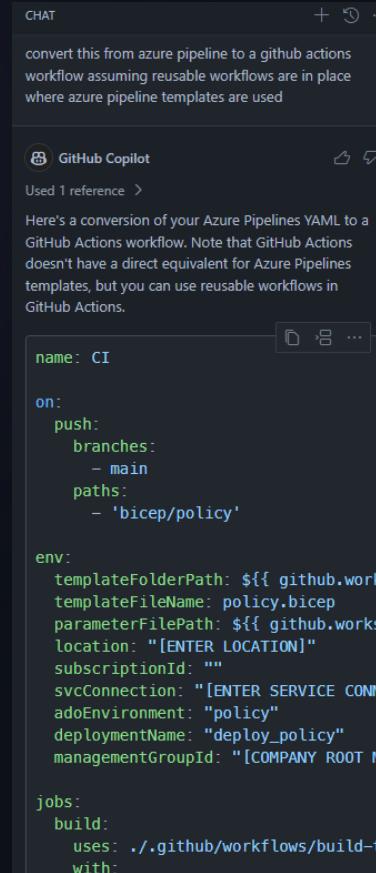
The screenshot shows a code editor interface with a dark theme. On the left, there is a sidebar titled "CHAT: GITHUB COPILOT" which says "Hi @bryantson, how can I help you?". It also includes a note: "I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve." The main area displays three Python scripts: "sorting-algorithm.py", "SimpleDemo.py", and "BubbleSort.py". The "sorting-algorithm.py" file contains implementations of quickSort, partition, bubbleSort, and insertionSort. The "SimpleDemo.py" file contains a quickSort call. The "BubbleSort.py" file contains a bubbleSort implementation. The status bar at the bottom right shows "Ln 19, Col 4" and "Python 3.11.5 64-bit".

```
sorting-algorithm.py 2: X
SimpleDemo > sorting-algorithm.py ...
1 def quickSort(array, low, high):
2     if low < high:
3         pivot = partition(array, low, high)
4         quickSort(array, low, pivot - 1)
5         quickSort(array, pivot + 1, high)
6     return array
7
8 def partition(array, low, high):
9     pivot = array[high]
10    i = low - 1
11    for j in range(low, high):
12        if array[j] < pivot:
13            i += 1
14            array[i], array[j] = array[j], array[i]
15    array[i + 1], array[high] = array[high], array[i + 1]
16    return i + 1
17
18 def bubbleSort(array) {
19     var len = array.length;
20     for (var i = 0; i < len; i++) {
21         for (var j = 0; j < len - i - 1; j++) {
22             if (array[j] > array[j + 1]) {
23                 var temp = array[j];
24                 array[j] = array[j + 1];
25                 array[j + 1] = temp;
26             }
27         }
28     }
29     return array;
30 }
31
32 def insertionSort(array) {
33     var len = array.length;
34     var preIndex, current;
35     for (var i = 1; i < len; i++) {
36         preIndex = i - 1;
37         current = array[i];
38         while (preIndex >= 0 && array[preIndex] > current) {
39             array[preIndex + 1] = array[preIndex];
40             preIndex--;
41         }
42         array[preIndex + 1] = current;
43     }
44     return array;

```

Convert code between languages

Convert code from
one programming
language to another



The screenshot shows a GitHub Copilot interface. At the top, there's a 'CHAT' bar with a message: 'convert this from azure pipeline to a github actions workflow assuming reusable workflows are in place where azure pipeline templates are used'. Below this, a GitHub Copilot card displays the converted code. The code is a GitHub Actions workflow named 'CI' that triggers on push events to the 'main' branch, specifically targeting the 'bicep/policy' directory. It uses a template from 'policy.bicep' and specifies a parameter file 'parameterFilePath'. The environment variables include 'templateFolderPath', 'templateFileName', 'parameterFilePath', 'location', 'subscriptionId', 'svcConnection', 'adoEnvironment', 'deploymentName', and 'managementGroupId'. The 'jobs' section contains a single 'build' job that runs a script from a GitHub workflow named 'build-t'.

```
name: CI

on:
  push:
    branches:
      - main
    paths:
      - 'bicep/policy'

env:
  templateFolderPath: ${{ github.work
  templateFileName: policy.bicep
  parameterFilePath: ${{ github.works
  location: "[ENTER LOCATION]"
  subscriptionId: ""
  svcConnection: "[ENTER SERVICE CONN"
  adoEnvironment: "policy"
  deploymentName: "deploy_policy"
  managementGroupId: "[COMPANY ROOT M

jobs:
  build:
    uses: ./github/workflows/build-t
    with:
```

Wiki markdown page generation

Making it easier
to document
best practices,
guidelines, and
technical
references.

The screenshot shows a dark-themed code editor interface. On the left, there's a sidebar with a "SOURCE CONTROL" section containing a message input field ("Message (Ctrl+Enter to commit on 'main')") and a "Commit" button. Below that is a "SOURCE CONTROL GRAPH" section showing a list of commits:

- Renamed README to Pages Dan Hellem (main)
- Added Authentication and Role-Based Access Control D...
- Added Data Modeling and Storage Dan Hellem
- Added API Design and Integration Dan Hellem
- Updated Frontend Architecture Dan Hellem
- Added Frontend Architecture Dan Hellem
- Updated Design System & UI Standards Dan Hellem
- Added Design System & UI Standards Dan Hellem
- Added README.md Dan Hellem

The main area displays a file named "API-Design-and-Integration.md" with the following content:

```
## API Design & Integration
...
### REST API Guidelines
- Use RESTful API design with resource-based endpoints.
- Implement authentication via JWT.
- Handle errors with meaningful response codes.

### Example API Service Wrapper
```javascript
import axios from 'axios';

const api = axios.create({
 baseURL: 'https://api.tractor-fleet.com/',
 headers: {
 'Content-Type': 'application/json',
 },
});

export const getTractors = async () => {
 try {
 const response = await api.get('/tractors');
 return response.data;
 } catch (error) {
 console.error('Error fetching tractors:', error);
 throw error;
 }
};
```
Ln 28, Col 4 (654 selected)
```

MCP Server Integration

- [AWS KB Retrieval](#) - Retrieval from AWS Knowledge Base using Bedrock Agent Runtime
- [Brave Search](#) - Web and local search using Brave's Search API
- [EverArt](#) - AI image generation using various models
- [Everything](#) - Reference / test server with prompts, resources, and tools
- [Fetch](#) - Web content fetching and conversion for efficient LLM usage
- [Filesystem](#) - Secure file operations with configurable access controls
- [Git](#) - Tools to read, search, and manipulate Git repositories
- [GitHub](#) - Repository management, file operations, and GitHub API integration
- [GitLab](#) - GitLab API, enabling project management
- [Google Drive](#) - File access and search capabilities for Google Drive
- [Google Maps](#) - Location services, directions, and place details
- [Memory](#) - Knowledge graph-based persistent memory system
- [PostgreSQL](#) - Read-only database access with schema inspection
- [Puppeteer](#) - Browser automation and web scraping
- [Redis](#) - Interact with Redis key-value stores
- [Sentry](#) - Retrieving and analyzing issues from Sentry.io
- [Sequential Thinking](#) - Dynamic and reflective problem-solving through thought sequences
- [Slack](#) - Channel management and messaging capabilities
- [Sqlite](#) - Database interaction and business intelligence capabilities
- [Time](#) - Time and timezone conversion capabilities

List all the app services in my sub c1a6ecdc-2290-4a0c-b941-f065b2c68131

I'll list all the App Services in your subscription.

- > Learning about appservice capabilities... - Azure MCP (MCP Server)
- > Learning about appservice capabilities... - Azure MCP (MCP Server)
- Let me use the Azure resource query tool to list the App Services:
- > Using "@azure query azure resource graph"

There is 1 App Service in subscription c1a6ecdc-2290-4a0c-b941-f065b2c68131:

- * arinco-crew-nearby (microsoft.web/sites) - Location: Australia Southeast

Find unused resources and provide script to remove them

I'll help you find unused Azure resources in your subscription and provide a script to remove them.

- > Using "@azure query azure resource graph"

I can see you have 201 resources in the subscription. Let me analyze different resource types to identify potentially unused resources:

- > Using "@azure query azure resource graph"

Perfect! I found several unused resources. Let me create a comprehensive PowerShell script to remove them:

✓ Created ✎ Remove-UnusedAzureResources.ps1

Unused Resources Found

I've identified 7 unused resources in your subscription and created a cleanup script:

- Deallocated Virtual Machines (3)
 - mcndc01 - mcn-ad-recovery-test-rg (deallocated since July 2024)
 - mcndc02 - mcn-ad-recovery-test-rg (deallocated since July 2024)
 - mcnwork01 - mcn-ad-recovery-test-rg (deallocated since July 2024)
- Unattached Network Interfaces (3)
 - dl-bricks-ui-pe-nic - dl-dev-rg
 - kube-apiserver.rnc24563277-0a9a-4183-8b02-9c5e601f6f9b - hb-ae-test-truckapi-nodes-rg
 - tor-ae-ai-ifc-account-pe-nic - tor-ae-ai-rg

> 1 file changed +201 -1

Keep Undo

mp.json +

Describe what to build next

Agent ✓ Claude Sonnet 4.5 ✓

All MCP servers 55

 **Markdown** Install

Convert various file formats (PDF, Word, Excel, images, audio) to Markdown.

By microsoft ⭐ 84,142

 **Netdata** Install

AI-powered infrastructure monitoring with real-time metrics, logs, alerts, and ML anomaly detection.

By netdata ⭐ 76,975

 **Context7** Install

Up-to-date code docs for any prompt

By upstash ⭐ 39,120

 **GitHub** Install

Connect AI assistants to GitHub - manage repos, issues, PRs, and workflows through natural language.

By github ⭐ 25,099

 **Playwright** Install

Automate web browsers using accessibility trees for testing and data extraction.

By microsoft ⭐ 24,241

 **Serena** Install

Semantic code retrieval & editing tools for coding agents.

By oralois ⭐ 17,057

 **ChromeDevTools** Install

MCP server for Chrome DevTools

By ChromeDevTools ⭐ 16,350

 **Firecrawl** Install

Extract web data with Firecrawl

By firecrawl ⭐ 5,051

 **Unity** Install

Control the Unity Editor from MCP clients via a Unity bridge + local Python server.

By CoplayDev ⭐ 4,227

 **Notion** Install

Official MCP server for Notion API

By makenotion ⭐ 3,573

 **Supabase** Install

MCP server for interacting with the Supabase platform

By supabase-community ⭐ 2,318

 **Azure MCP Server** Install

All Azure MCP tools to create a seamless connection between AI agents and Azure services.

By microsoft ⭐ 2,285

 **Stripe** Install

 **Microsoft Learn** Install

 **Terraform** Install

Common Problems

Problems #1:

Copilot fails to produce answer or keep repeating

Some problems

- Fails to produce answer
- Hallucination - Keeps repeating

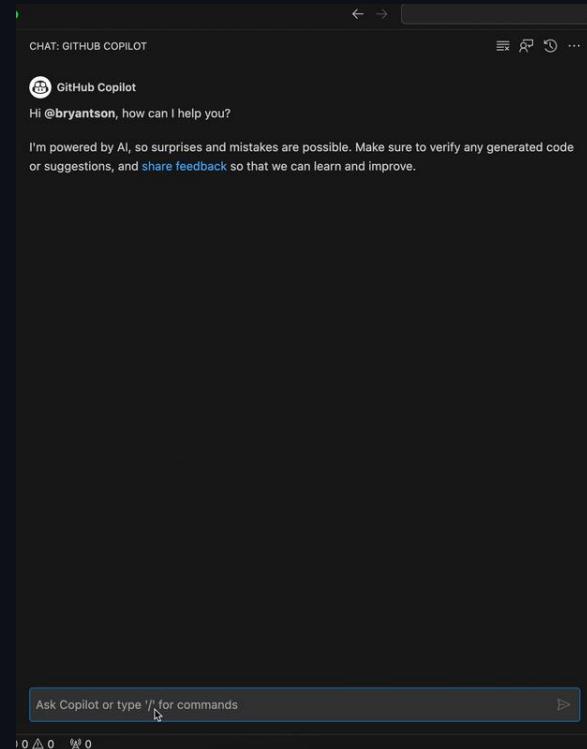
A screenshot of a dark-themed code editor interface. On the left, the Explorer sidebar shows a workspace named 'SimpleDemo' containing a file named 'calculator.py'. The main area displays the first line of 'calculator.py': '1 |Press ⌘ + ⌂ to ask GitHub Copilot Chat to do something. Start typing to dismiss.' The status bar at the bottom right indicates the current line and column as 'Ln 1, Col 1', and shows other details like 'Spaces: 4', 'UTF-8', 'LF', 'Python 3.11.5 64-bit', and a refresh icon.

Problems #2:

Copilot generates incorrect result

Reasons why...

- Not enough context
- Old trained data
- Generative in nature

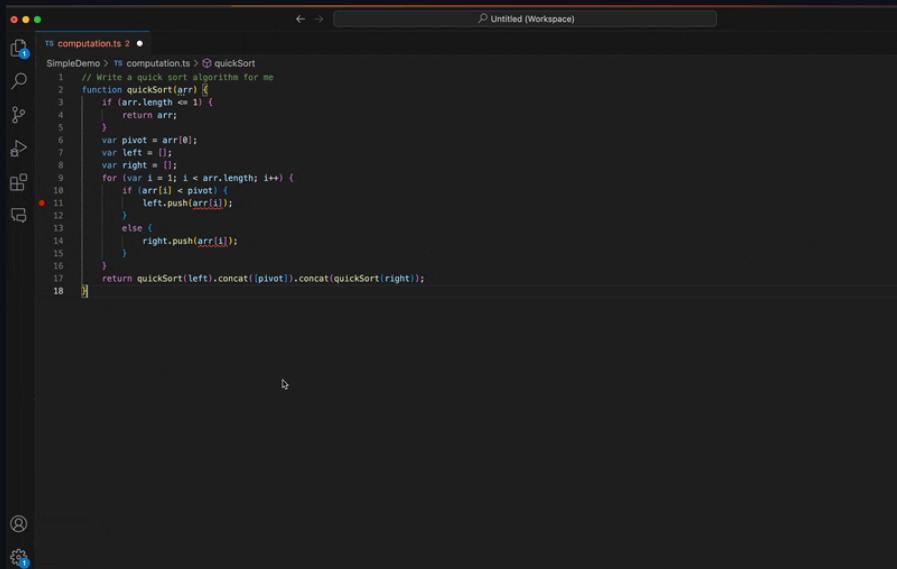


Problem #3:

Copilot suggests non-optimal solution

How to improve

- Ask in Copilot Chat
- Use in-line suggestion from Copilot



A screenshot of a code editor window titled "TS computation.ts". The code is a quicksort algorithm. A red dot on the status bar indicates a warning or error. The code is as follows:

```
SimpleDemo > ts computation.ts ⚡ quickSort
1 // Write a quick sort algorithm for me
2 function quickSort(arr) {
3     if (arr.length <= 1) {
4         return arr;
5     }
6     var pivot = arr[0];
7     var left = [];
8     var right = [];
9     for (var i = 1; i < arr.length; i++) {
10        if (arr[i] < pivot) {
11            left.push(arr[i]);
12        }
13        else {
14            right.push(arr[i]);
15        }
16    }
17    return quickSort(left).concat([pivot]).concat(quickSort(right));
18 }
```

Q & A

Hands-on Exercise

- Terraform
- <https://github.com/shin-akuma/copilot-terraform>
- Scripting
- <https://experience.cloudlabs.ai/#/labguidepreview/f9fd80ac-fc1a-4609-a3f1-06650aec389e> (Page 10)
- <https://github.com/shin-akuma/copilot-scripting>

Upcoming Sessions

1

Q&A Session



Thank you