**Suggestions for Standard Resource Structure**

**What Copilot does:** When you start writing a resource, variable, provider, or output block, Copilot detects the pattern and automatically suggests a complete structure based on best practices or official documentation.

**Benefits:**

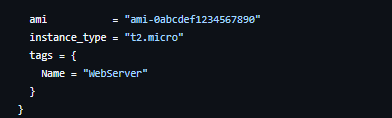
* Saves time writing resources from scratch
* Reduces syntax errors
* Acts as a learning aid for beginners

Example:

If you write:



Copilot might suggest:



## 

## **More Descriptive and Consistent Variable Names**

**What Copilot does:** It recommends variable, resource, and output names based on project context and naming best practices. This improves readability and maintainability.

**Benefits:**

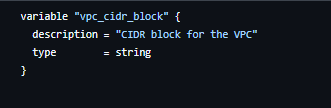
* Promotes team-wide consistency
* Avoids generic or confusing names
* Improves documentation of code intent

**Example:**

Instead of writing:



Copilot may suggest:



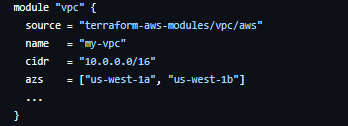
## **Quick Generation of Reusable Module Blocks**

**What Copilot does:** When you start writing a module block, Copilot suggests commonly used parameters. For well-known modules (like those from the Terraform Registry), Copilot often “knows” the expected structure.

**Benefits:**

* Speeds up integration of complex modules
* Minimizes mistakes with incorrect inputs
* Encourages reusable infrastructure-as-code

**Example:**



Copilot can auto-complete this based on the module source.

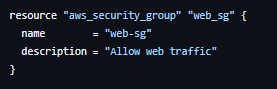
## **Automatic Generation of Helpful Comments**

**What Copilot does:** It detects your code and suggests natural language comments explaining what the block does.

**Benefits:**

* Enhances in-code documentation
* Helps collaborators understand and maintain infrastructure
* Great for multi-team or onboarding scenarios

**Example:**



These comments can be automatically suggested by Copilot.

## 

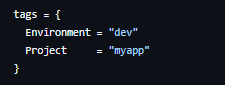
## **Simplifying Repetitive Patterns and Refactoring Suggestions**

**What Copilot does:** It detects code repetition and may suggest more efficient ways to handle it, such as using locals, count, for\_each, or extracting values into variables.

**Benefits:**

* Reduces code duplication
* Encourages clean architecture
* Improves scalability and reusability

**Before Example:**



Copilot may suggest defining this in locals:



And then reusing it:

