Click Here

Get-Process -Id (Get-NetTCPConnection -LocalPort YourPortNumberHere).OwningProcess shows several columns of information about a process. The Id column is the PID you need if you want to kill it with taskkill /PID . UDP Get-Process -Id (Get-NetUDPEndpoint -LocalPort YourPortNumberHere).OwningProcess is useful for finding information about a specific port. netstat -a -b displays all connections and listening ports, including the executable involved in creating each connection or listening port.Note: Dane's recommendation for TCPView is to use it because it looks very useful! -a Displays all connections and listening ports, which can be time-consuming if you don't have sufficient permissions. In some cases well-known executables host multiple independent components, and in these cases the sequence of components involved in creating the connection or listening port is displayed.There's a native GUI for Windows: Resource Monitor (Start menu All Programs Accessories System Tools) or run resmon.exe, or from TaskManager Performance tab. For Windows: netstat -aon | find /i "listening".Note that the -b switch requires administrative privileges on the machine. You don't really need elevated rights to get the process name! Find the pid of the process running in the port number (e.g., 8080) by using the 'Find' command: netstat -ano | findStr "8080" or tasklist /fi "pid eq 2216".You can get more information if you run the following command: netstat -aon | find /i "listening" |find "port". Using the 'Find' command allows you to filter the results.Open a command prompt window (as Administrator) and enter the text then hit Enter. netstat -abno -a Displays all connections and listening ports, including the executable involved in creating each connection or listening port.Note: To find the process under Task Manager, look at the PID (process identifier) next to the port you are looking at. Open Windows Task Manager, select the Processes tab, look for the PID, and make sure Show processes from all users is selected.To identify a process using a specific port in Windows, follow these steps:After executing the command `taskkill /F /PID `, where `` is the last column of the output, select the desired process by right-clicking on Command Prompt and running it as administrator. Alternatively, use PowerShell to run the `Get-NetTCPConnection` cmdlet.Format the output to include only the properties you're interested in, such as LocalAddress, LocalPort, State, and OwningProcess. To get a table with these properties, use the following command:PS C:\> Get-NetTCPConnection -LocalPort | Format-Table -Property LocalAddress, LocalPort, State, OwningProcessIf you want to find the name of the process using a specific port, run the following command:PS C:\> Get-Process -Id (Get-NetTCPConnection -LocalPort).OwningProcessTo get a list of all owning process IDs associated with each connection, use the `netstat -ao | find /i "listening"` command.If you want to kill any process, have the ID and use the following command: Taskkill /F /IM PID of a process.To find the application using your port number, type `netstat -aon | findstr :DESIRED_PORT_NUMBER` into the command line. For instance, to locate port 80, use `netstat -aon | findstr :80`. The answer was previously posted in this question, which showed how to identify processes running on specific ports.The `netstat -ao` and `netstat -ab` commands can reveal application names, but you may face an issue if you're not a system administrator, resulting in the "The requested operation requires elevation" message. Fortunately, Sysinternals' Process Explorer can help by accessing the TCP tab to verify whether the desired process is using that port.For PowerShell users, try `Get-NetworkStatistics` to filter network statistics and find processes running on specific ports. For example:```> Get-NetworkStatistics | where Localport -eq 8000ComputerName : DESKTOP-JL59SC6Protocol : TCPLocalAddress : 0.0.0.0LocalPort : 8000RemoteAddress : 0.0.0.0RemotePort : 0State : LISTENINGProcessName : nodePID : 11552```Another recommended tool is CurrPorts from NirSoft, which can filter displayed results and help identify processes using specific ports.To close a process running on a particular port in Windows, you can use the `netstat -a -o` command to obtain the process ID (PID), then end that process in Task Manager. You can also kill a process remotely using the `taskkill /pid pid /f` command, where `pid` is the desired process ID.For more advanced network analysis and process identification, consider tools like Process Explorer, Port Monitor, or wmic.Additionally, you can use the `netsh int ipv4 show excludedportrange protocol=tcp` command to view reserved ports on your system.Using Git to Identify the Best Candidate for a Unique IdentifierWe can see that classmethod and staticmethod are Non-Data-Descriptors, as they only have the __get__ method. Similarly, all functions are also Non-Data-Descriptors. However, property is a Data-Descriptor, as it has the __set__, __get__, and __delete__ methods.In dotted lookup order, Python first checks if the attribute is a Data-Descriptor on the class of the instance. If not, it looks in the instance's __dict__ and finally falls back to Non-Data-Descriptors. This means that Non-Data-Descriptors like functions/methods can be overridden by instances.To ensure we always have a float for this class attribute of Temperature, and that we can't use del to delete the attribute, we need a descriptor class. The descriptor class stores state in itself, instead of the owner-class and instances of the owner, which is why it's necessary.We could just as easily share state across all instances with a simple class attribute, but this would require setting it as a float for the class and never deleting it, or being comfortable with users of our code doing so. However, using Python's built-in property decorator achieves the same behavior in a more idiomatic way.The attributes of descriptors in Python are defined by the way you use variables to represent data types, especially for complex objects. In addition, there are two main kinds: data-descriptors and non-data-descriptors. Builtin objects like lists, dictionaries, and sets utilize these descriptors to define their behavior when used with attributes. When using descriptors, it's often best to first consider a simple class attribute method, moving on only if necessary to use the property decorator for further control.Now lets move on to HTTP Verbs: POST, GET, PUT, DELETE and PATCH.We can use Fiddler for checking the response. Open Fiddler and select the Compose tab. Specify the verb and url as shown below and click Execute to check the response. Verb: DELETE url: Response: On successful deletion it returns HTTP status 200 (OK) along with a response body. Example between PUT and PATCH PUT If I had to change my first name then send PUT request for Update: { "first": "Nazmul", "last": "hasan" } So, here in order to update the first name we need to send all the parameters of the data again. PATCH: Patch request says that we would only send the data that we need to modify without modifying or affecting other parts of the data. Ex: if we need to update only the first name, we pass only the first name. Please refer the below links for more information: docker attach will let you connect to your Docker container, but this isn't really the same thing as ssh. If your container is running a webserver, for example, docker attach will probably connect you to the stdout of the web server process. It won't necessarily give you a shell. The docker exec command is probably what you are looking for; this will let you run arbitrary commands inside an existing container. For example, to run bash inside a container: docker exec -it sh Of course, whatever command you are running must exist in the container filesystem; if your container doesn't have sh, this will fail with something like: OCI runtime exec failed: exec failed: unable to start container process: exec: "sh": executable file not found in $PATH: unknown [If your container doesn't have sh -- which is a common case for minimal images -- you may need to investigate other ways to explore the container filesystem.] In the above command is the name or ID of the target container. It doesn't matter whether or not you're using docker compose; just run docker ps and use either the ID (a hexadecimal string displayed in the first column) or the name (displayed in the final column). E.g., given: $ docker ps d2d4a89aaee9 larsks/mini-httpd "mini_httpd -d /cont 7 days ago Up 7 days web I can run: $ docker exec -it web ip addr 1: lo: mtu 65536 qdisc noqueue state UNKNOWN link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 scope host valid_lft forever preferred_lft forever 18: eth0: mtu 1500 qdisc noqueue state UP link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff inet 172.17.0.3/16 scope global eth0 valid_lft forever preferred_lft forever inet6 fe80::42:acff:fe11:3/64 scope link valid_lft forever preferred_lft forever I could accomplish the same thing by running: $ docker exec -it d2d4a89aaee9 ip addr Similarly, I could start a shell in the container; $ docker exec -it web sh / # echo This is inside the container. This is inside the container. / # exit $ In commands shown in this answer, the -i and -t options (combined as -it) are necessary to get an interactive shell: -i keeps stdin connected; if you don't specify -i, the shell will simply exit. -t allocates a tty device; if you don't specify -t, you won't have a very pleasant interactive experience (there will be no shell prompt or job control, for example). If you're specifically using docker compose, there is a convenience docker compose exec command that works very much like the docker exec command, except: It defaults to the behavior of -i and -t It allows you to refer to containers by their service name in your compose.yaml file. For example, if you have a compose.yaml like this: services: web: image: docker.io/alpinelinux/darkhttpd Then you can run: docker compose exec web sh The equivalent docker exec command would be something like: docker exec -it myproject-web-1 sh Page 2 I'm getting started working with Docker. I'm using the WordPress base image and docker-compose. I'm trying to ssh into one of the containers to inspect the files/directories that were created during the initial build. I tried to run docker-compose run containername ls -la, but that didn't do anything. Even if it did, I'd rather have a console where I can traverse the directory structure, rather than run a single command. What is the right way to do this with Docker? There is actually a default pattern that you can employ to achieve this result without having to implement IDesignTimeDbContextFactory and do any config file copying.###To instantiate your DbContext at design time, you need to leverage the new hook `public static IWebHost BuildWebHost(string[] args)`. This method can be implemented in any class that houses your entry point. The framework uses this hook to connect your configuration and create a default pattern for new projects.The process starts with adding a migration. When the framework attempts to create your DbContext, it looks for IDesignTimeDbContextFactory implementations in a collection of factory methods. It then gets your configured services via the static hook discussed earlier and adds any context types registered with DbContextOptions in the Startup.ConfigureServices method.The framework uses an order of precedence to resolve this process:1. IDesignTimeDbContextFactory implementations are used first.2. The static hook is used next.3. Any context types registered with DbContextOptions are added after that.For most common scenarios, you won't need to implement IDesignTimeDbContextFactory. Instead, focus on implementing the static hook in your entry point class.###ARTICLETo terminate a process using the port number, you can follow Solution 1: Kill Process Run command-line as an Administrator netstat -ano | findstr : taskkill /PID /FAlternatively, you can change the port by making sure it doesn't listen to any other process. Solution 2: Change PortMake sure that new port you are going to set for your Application doesn't listen to any other process.You can check if a port is in use using Solution 3:Another way is to terminate the process (in IDE) and clean and rebuild project.To check port status, you can use Option 1: Run resmon.exe and go to Network -> Listening Port. Alternatively, you can use PowerShell with Get-Process and netstat commands.

How to get tampermonkey. How to use tampermonkey scripts. Tampermonkey tutorial. Tampermonkey how to run script. How to get tampermonkey to work on chrome. How to enable tampermonkey scripts. Why is tampermonkey not working. How to get tampermonkey on ios.