| opentelemetry-bootstrap | This commands inspects the active Python site-packages and figures out which instrumentation packages the user might want to install. | 1. 1 2. opentelemetry-bootstrap -a install |
| --- | --- | --- |
| opentelemetry-distro | They include a means to automatically set some of the more popular options for users in order to make using OpenTelemetry and auto-instrumentation as quick as possible without compromising flexibility. | 1. 1 2. pip install opentelemetry-distro |
| opentelemetry-instrument | The instrument command will try to automatically detect packages used by your python program and when possible, apply automatic tracing instrumentation on them. | 1. 1 2. 2 3. 3 4. 4 5. 5 6. opentelemetry-instrument \ 7. --traces\_exporter console \ 8. --metrics\_exporter console \ 9. --logs\_exporter console \ 10. flask run -p 8080 |
| pip | To make sure that requests will function, the pip programme searches for the package in the Python Package Index (PyPI), resolves any dependencies, and installs everything in your current Python environment. | 1. 1 2. pip list |

| pip install | The pip install <package> command looks for the latest version of the package and installs it. | pip install example\_package |
| --- | --- | --- |
| source virtualenv | Activate the virtual environment | 1. 1 2. source myenv/bin/activate |

| touch | It can create multiple files and updates the modified time if files exist. | 1. 1 2. touch [OPTIONS] FILENAME |
| --- | --- | --- |
| virtualenv | It is primarily a command line application, that modifies the environment variables in a shell to create an isolated Python environment. | 1. 1 2. 2 3. pip install virtualenv 4. virtualenv myenv |

| **ackage/Method** | **Description** | **Code Example** |
| --- | --- | --- |
| bash | Bash, or the Bourne Again Shell command, is a command-line interpreter commonly used in Unix-based operating systems. It runs in a text window where the user can interpret commands to carry out various actions. | 1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. Example: This generates a list of numbers and prints them: 8. #!/bin/bash 9. # Loop from 1 to 3 and print the numbers 10. for i in {1..3}; do 11. echo "Number: $i" 12. done |