

## Variant 1

---

1. In a Dockerfile, which instruction sets the base image for a build?

- a) IMAGE
- b) BASE
- c) FROM
- d) PARENT

2. Which command lists all containers, including stopped ones?

- a) `docker ps`
- b) `docker list all`
- c) `docker ps -a`
- d) `docker containers --all`

3. In 'unittest', what class must your test classes inherit from?

- a) `unittest.Test`
- b) `unittest.TestCase`
- c) `unittest.Suite`
- d) `unittest.Framework`

4. What is the most direct command to run all discovered tests using 'pytest'?

- a) `run pytest`
- b) `python -m pytest.run`
- c) `pytest`
- d) `pytest --all`

5. Which command runs tests from a single file with 'unittest'?

- a) `unittest run test_file.py`
- b) `python test_file.py`
- c) `python -m unittest test_file.py`
- d) `run unittest test_file.py`

## Variant 2

---

1. In a Dockerfile, which instruction is preferred for copying local files into the container?

- a) PUT
- b) COPY
- c) MOVE
- d) INSERT

2. How do you map a host port 8080 to a container port 80?

- a) `docker run -p 8080:80`
- b) `docker run --port 8080-80`
- c) `docker run -m 8080:80`
- d) `docker run --map 8080/80`

3. In 'pytest', how are test functions typically named?

- a) `function_test()`
- b) `TestMyFunction()`
- c) `test_my_function()`
- d) `run_test_my_function()`

4. Which is the current standard 'unittest' method to check for equality?

- a) `self.checkEquals()`
- b) `self.isEqual()`
- c) `self.assert_equals()`
- d) `self.assertEqual()`

5. In 'pytest', what is the standard way to write a test assertion?

- a) `check(x == y)`
- b) `assertEqual(x, y)`
- c) `assert x == y`
- d) `verify(x, y)`

## Variant 3

---

1. Which command shows the logs of a container named 'webapp'?

- a) `docker info webapp`
- b) `docker attach webapp`
- c) `docker logs webapp`
- d) `docker history webapp`

2. In a Dockerfile, which instruction executes a command during the build?

- a) CMD
- b) EXEC
- c) RUN
- d) ENTRYPOINT

3. Which 'unittest' methods run before and after each test method?

- a) `start()` / `stop()`
- b) `init()` / `destroy()`
- c) `setUp()` / `tearDown()`
- d) `begin()` / `end()`

4. How does 'pytest' discover test files by default?

- a) Files named `suite_*.py`
- b) Files named `test_*.py` or `*_test.py`
- c) All files ending in `.py`
- d) Files listed in a `pytest.conf` file

5. How would you run 'pytest' with verbose output?

- a) `pytest --show-all`
- b) `pytest -v`
- c) `pytest --debug`
- d) `pytest -d`