

## Get started

Welcome

Installation >

Running k6

Results output

k6 resources

## Using k6

HTTP Requests

Metrics >

Checks

Thresholds

Options >

Test lifecycle

k6.io

app.k6.io

Search

[Using k6](#) > [Scenarios](#) > [Executors](#) > [Per VU iterations](#)

## Per VU iterations

 [SUGGEST EDITS](#)

With the `per-vu-iterations` executor, each VU executes an exact number of iterations. The total number of completed iterations equals `vus * iterations`.

[Options](#)[When to use](#)[Example](#)[Observations](#)

### Options

Besides the [common configuration options](#), this executor has the following options:

OPTION	TYPE	DESCRIPTION	DEFAULT
<code>vus</code>	integer	Number of VUs to run concurrently.	<code>1</code>
<code>iterations</code>	integer	Number of <code>exec</code> function iterations to be executed by each VU.	<code>1</code>
<code>maxDuration</code>	string	Maximum scenario duration before it's forcibly stopped (excluding <code>gracefulStop</code> ).	<code>"10m"</code>

### When to use

Use this executor if you need a specific number of VUs to complete the same number of iterations. This can be useful when you have fixed sets of test data that you want to partition between VUs.

### Example

The following example schedules 10 VUs to execute 20 iterations *each*. The test runs 200 total iterations and has a maximum duration of 30 seconds.

**per-vu-its.js**

```
1 import http from 'k6/http';
2 import { sleep } from 'k6';
3
4 export const options = {
5   discardResponseBodies: true,
6   scenarios: {
7     contacts: {
8       executor: 'per-vu-iterations',
9       vus: 10,
10      iterations: 20,
11      maxDuration: '30s',
12    },
13  },
14 };
15
16 export default function () {
17   http.get('https://test.k6.io/contacts.php');
18   // Injecting sleep
19   // Sleep time is 500ms. Total iteration time is sleep + time to finish
20   sleep(0.5);
21 }
```

### Observations

The following graph depicts the performance of the [example](#) script:





Based upon our test scenario inputs and results:

- The number of VUs is fixed at 10, and are initialized before the test begins;
- Total iterations are fixed at 20 iterations per VU, i.e. 200 iterations,  $10 \text{ VUs} * 20 \text{ iters each}$ ;
- Each *iteration* of the default function is expected to be roughly 515ms, or  $\sim 2/s$ ;
- Maximum throughput (highest efficiency) is therefore expected to be  $\sim 20 \text{ iters/s}$ ,  $2 \text{ iters/s} * 10 \text{ VUs}$ ;
- The maximum throughput is reached, but not maintained;
- Because the distribution of iterations is even among VUs, a *fast* VU may finish early and be idle for the remainder of the test, thereby lowering *efficiency*;
- Total duration of 9 seconds is slightly longer than *shared iterations* due to lower efficiency;
- Overall test duration lasts as long as the *slowest* VU takes to complete 20 requests.

< [PREVIOUS](#)  
Shared iterations

[NEXT](#) >  
Constant VUs



#### PRODUCT

Open Source  
Grafana Cloud k6  
Grafana Cloud k6 Pricing  
Open Source vs Cloud  
Build vs Buy  
Testimonials

#### RESOURCES

k6 Docs  
Grafana Cloud k6 Docs  
Extensions  
Integrations  
Modern Load Testing  
Not a developer. Why k6?

#### COMMUNITY

Engage♥  
Forum  
Slack  
GitHub  
k6 Champions

#### ABOUT

Blog  
Our story  
Our beliefs  
Contact  
Jobs🔗

#### Subscribe to our newsletter!

Product developments and news from the k6 community.

SUBSCRIBE NOW