k6 DOCS Grafana Labs

Guides

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

GUIDES JAVASCRIPT API CLOUD DOCS EXTENSIONS INTEGRATIONS EXAMPLES

Search

Using k6 > Scenarios > Executors > Constant arrival rate

Constant arrival rate

SUGGEST EDITS

With the `constant-arrival-rate` executor, k6 starts a fixed number of iterations over a specified period of time. It is an open-model executor, meaning iterations start independently of system response (for details, read Open and Closed models).

This executor continues to start iterations at the given rate as long as VUs are available. The time to execute an iteration can vary with test logic or the system-under-test response time. To compensate for this, the executor starts a varied number of VUs to meet the configured iteration rate. For explanations of how allocation works, read Arrival-rate VU allocation.

It is considered an open model because it does not impose any restrictions on the number of arrivals or the capacity of the system to handle incoming requests. New arrivals are allowed to join the system at any time, even if the system is already busy processing previous requests.

NOTE

Iteration starts are spaced fractionally.

Iterations **do not** start at exactly the same time. At a `rate` of `10` with a `timeUnit` of `1s`, each iteration starts about every tenth of a second (that is, each 100ms).

Options

Besides the common configuration options, this executor has the following options:

OPTION	TYPE	DESCRIPTION	DEFAULT
duration(required)	string	Total scenario duration (excluding `gracefulStop`).	-
rate(required)	integer	Number of iterations to start during each `timeUnit` period.	-
preAllocatedVUs(required)	integer	Number of VUs to pre-allocate before test start to preserve runtime resources.	-
timeUnit	string	Period of time to apply the `rate` value.	`"1s"`
maxVUs	integer	Maximum number of VUs to allow during the test run.	If unset, same as `preAllocatedVUs`

When to use

When you want iterations to remain constant, independent of the performance of the system under test. This approach is useful for a more accurate representation of RPS, for example.

NOTE

Don't put sleep at the end of an iteration.

The arrival-rate executors already pace the iteration rate through the `rate` and `timeUnit` properties. So it's unnecessary to use a `sleep()` function at the end of the VU code.

Example

This example schedules a constant rate of 30 iterations per second for 30 seconds. It allocates 50 VUs for k6 to dynamically use as needed.

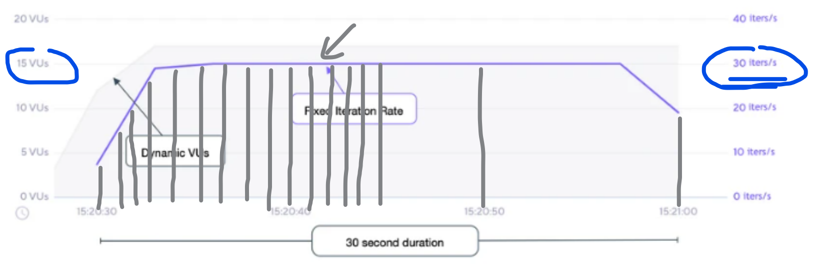
constant-arr-rate.js

```javascript
import http from 'k6/http';

export const options = {
  discardResponseBodies: true,
  scenarios: {
    contacts: {
      executor: 'constant-arrival-rate',

      // How long the test lasts
      duration: '30s',

      // How many iterations per timeUnit
      rate: 30,

      // Start `rate` iterations per second
      timeUnit: '1s',

      // Pre-allocate VUs
      preAllocatedVUs: 50,
    },
  },
};

export default function () {
  http.get('https://test.k6.io/contacts.php');
}
```

Constant rate 30 iters/s

Observations

The following graph depicts the performance of the **example** script:



Based upon our test scenario inputs and results:

- The desired rate of 30 iterations started every 1 second is achieved and maintained for the majority of the test.
- The test scenario runs for the specified 30 second duration.
- Having started with 2 VUs (as specified by the `preAllocatedVUs` option), k6 automatically adjusts the number of VUs to achieve the desired rate, up to the allocated number. For this test, this ended up as 17 VUs.
- Exactly 900 iterations are started in total, `30s * 30 iters/s`.

> Using too low of a `preAllocatedVUs` setting will reduce the test duration at the desired rate, as resources need to continually be allocated to achieve the rate.

PREVIOUS
Ramping VUs

NEXT
Ramping arrival rate

Grafana Cloud k6 Pricing	Extensions	Slack	Our beliefs
Open Source vs Cloud	Integrations	GitHub	Contact
Build vs Buy	Modern Load Testing	k6 Champions	Jobs 4
Testimonials	Not a developer. Why k6?		

Subscribe to our newsletter!
Product developments and news from the k6 community.

Email

SUBSCRIBE NOW