



# FrontLine User Guide

Gatling Corp

Version 1.12.1, 2020-09-29

# Table of Contents

1. Developer Guide .....	1
1.1. FrontLine Gatling Versions.....	1
1.2. Configuring Gatling Projects .....	1
1.2.1. Maven .....	1
1.2.2. SBT .....	3
1.2.3. Gradle.....	5
1.2.4. Multi-Module Support .....	5
1.3. Note on Feeders.....	5
1.4. Specific Gatling Features.....	6
1.4.1. Load Sharding .....	6
1.5. Resolving Injector Location in Simulation .....	6
1.6. Publishing Fatjars into Binary Repositories.....	7
1.6.1. Maven .....	7
1.6.2. Gradle.....	7
1.6.3. Sbt .....	8
2. Dashboard User Guide .....	9
2.1. Configuration.....	9
2.2. Login .....	9
2.3. Overview .....	10
2.4. Admin .....	11
2.4.1. Managing Teams .....	11
2.4.2. Managing Users.....	12
2.4.3. Managing API Tokens.....	14
2.4.4. Managing Private Keys .....	15
2.4.5. Managing Repositories .....	16
2.5. Accessing your profile .....	21
2.6. Pools.....	22
2.6.1. Warning about Private Keys .....	23
2.6.2. On-premises .....	23
2.6.3. AWS Pool .....	25
2.6.4. GCE Pool (On-premises license only) .....	27
2.6.5. OpenStack Pool (On-premises license only) .....	29
2.6.6. Microsoft Azure Pool (On-premises license and Azure Marketplace only) .....	31
2.6.7. Kubernetes/OpenShift Pool .....	33
2.7. Simulations .....	37
2.7.1. Global Properties .....	38
2.7.2. Creating a simulation .....	40
2.7.3. Simulations table .....	46

2.7.4. Useful tips . . . . .	47
2.7.5. Run / Trends . . . . .	48
2.7.6. Reports . . . . .	52
2.8. Documentation . . . . .	60
2.9. Plugins Download . . . . .	61
2.10. About . . . . .	61
3. Public APIs . . . . .	63

# Chapter 1. Developer Guide

## 1.1. FrontLine Gatling Versions

FrontLine actually uses custom versions of the Gatling components. Those binaries are not open sources and their usage is restricted to FrontLine. When you'll be deploying tests with FrontLine, it will replace your Gatling OSS dependencies with their custom counterparts.

## 1.2. Configuring Gatling Projects

### 1.2.1. Maven

In your `pom.xml`, you have to add in:

- pull Gatling dependencies
- add the maven plugin for Scala, so your code gets compiled
- add the maven plugin for FrontLine, so it can package your code into a deployable artifact

*pom.xml:*

```
<dependencies>
  <dependency>
    <groupId>io.gatling.highcharts</groupId>
    <artifactId>gatling-charts-highcharts</artifactId>
    <version>3.4.0</version>
    <scope>test</scope>
  </dependency>
</dependencies>

<build>
  <!-- so maven compiles src/test/scalar and not only src/test/java -->
  <testSourceDirectory>src/test/scalar</testSourceDirectory>
  <plugins>
    <plugin>
      <artifactId>maven-jar-plugin</artifactId>
      <version>3.2.0</version>
    </plugin>
    <!-- so maven can compile your scalar code -->
    <plugin>
      <groupId>net.alchim31.maven</groupId>
      <artifactId>scala-maven-plugin</artifactId>
      <version>{scalaMavenPluginVersion}</version>
      <executions>
        <execution>
          <goals>
            <goal>testCompile</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

```

<recompileMode>all</recompileMode>
<jvmArgs>
  <jvmArg>-Xss100M</jvmArg>
</jvmArgs>
<args>
  <arg>-target:jvm-1.8</arg>
  <arg>-deprecation</arg>
  <arg>-feature</arg>
  <arg>-unchecked</arg>
  <arg>-language:implicitConversions</arg>
  <arg>-language:postfixOps</arg>
</args>
</configuration>
</execution>
</executions>
</plugin>

<!-- so maven can build a package for FrontLine -->
<plugin>
  <groupId>io.gatling.frontline</groupId>
  <artifactId>frontline-maven-plugin</artifactId>
  <version>1.2.0</version>
  <executions>
    <execution>
      <goals>
        <goal>package</goal>
      </goals>
    </execution>
  </executions>
</plugin>
</plugins>
</build>

```

You can run `mvn package -DskipTests` in your terminal and check you get a jar containing all the dependencies of the simulation.

You can also exclude dependencies you don't want to ship, eg:

pom.xml:

```
<plugin>
  <groupId>io.gatling.frontline</groupId>
  <artifactId>frontline-maven-plugin</artifactId>
  <version>1.2.0</version>
  <executions>
    <execution>
      <goals>
        <goal>package</goal>
      </goals>
      <configuration>
        <excludes>
          <exclude>
            <groupId>org.scalatest</groupId>
            <artifactId>scalatest_2.12</artifactId>
          </exclude>
        </excludes>
      </configuration>
    </execution>
  </executions>
</plugin>
```

### 1.2.2. SBT

In a sbt project, you have to:

- pull Gatling dependencies
- add the sbt plugin for FrontLine, so it can package your code into a deployable artifact

A `build.sbt` file should look like this:

*build.sbt:*

```
enablePlugins(GatlingPlugin, FrontLinePlugin)

ThisBuild / Keys.useCoursier := false

// If you want to package simulations from the 'it' scope instead
inConfig(IntegrationTest)(_root_.io.gatling.frontline.sbt.FrontLinePlugin.frontlineSettings(IntegrationTest))

scalaVersion := "2.12.12"
scalacOptions := Seq(
  "-encoding", "UTF-8", "-target:jvm-1.8", "-deprecation",
  "-feature", "-unchecked", "-language:implicitConversions", "-language:postfixOps")

val gatlingVersion = "3.4.0"

libraryDependencies += "io.gatling.highcharts" % "gatling-charts-highcharts" % gatlingVersion % "test"
// only required if you intend to use the gatling-sbt plugin
libraryDependencies += "io.gatling"          % "gatling-test-framework"    % gatlingVersion % "test"
```



**Don't forget to replace the UUID with the one you were given.**



We only support sbt 1+, not sbt 0.13.



We recommend disabling Coursier for now. There are several bugs in the sbt/Coursier integration that makes our plugin work in a suboptimal fashion.

INFO: The `gatling-test-framework` dependencies is only needed if you intend to run locally and use the gatling-sbt plugin.

INFO: If you use very long method calls chains in your Gatling code, you might have to increase sbt's thread stack size:

```
$ export SBT_OPTS="-Xss100M"
```

You will also need the following lines in the [project/plugins.sbt](#) file:

*project/plugins.sbt*

```
// only if you intend to use the gatling-sbt plugin for running Gatling locally
addSbtPlugin("io.gatling" % "gatling-sbt" % "3.1.0")
// so sbt can build a package for FrontLine
addSbtPlugin("io.gatling.frontline" % "sbt-frontline" % "1.3.0")
```

You can run `sbt test:assembly` (or `sbt it:assembly` if you've configured the plugin for integration tests) in your terminal and check you get a jar containing all the dependencies of the simulation.

INFO: The `gatling-sbt` is optional.

### 1.2.3. Gradle

In a Gradle project, you have to:

- pull Gatling dependencies
- add the gradle plugin for FrontLine, so it can package your code into a deployable artifact

A `build.gradle` file should look like this:

*build.gradle:*

```
plugins {
    // The following line allows to load io.gatling.gradle plugin and directly apply
    it
    id 'io.gatling.frontline.gradle' version '1.2.0'
}

// This is needed to let io.gatling.gradle plugin to loads gatling as a dependency
repositories {
    jcenter()
    mavenCentral()
}

gatling {
    toolVersion = '3.4.0'
}
```



**Don't forget to replace the UUID with the one you were given.**

You can run `gradle frontlineJar` in your terminal and check you get a jar containing all the dependencies of the simulation.

### 1.2.4. Multi-Module Support

If your project is a multi-module one, make sure that only the one containing the Gatling Simulations gets configured with the Gatling related plugins describes above. FrontLine will take care of deploying all available jars so you can have Gatling module depend on the other ones.

## 1.3. Note on Feeders

A typical mistake with Gatling and FrontLine is to rely on having an exploded maven/gradle/sbt project structure and try loading files from the project filesystem.

This filesystem structure will be gone once FrontLine will have compiled your project and uploaded

your binaries on the injectors.

If your feeder files are packaged with your test sources, you must resolve them from the classpath. This way will always work, both locally and with FrontLine.

```
// incorrect  
val feeder = csv("src/test/resources/foo.csv")  
  
// correct  
val feeder = csv("foo.csv")
```

## 1.4. Specific Gatling Features

### 1.4.1. Load Sharding

Injection rates and throttling rates are automatically distributed amongst nodes.

However, Feeders data is not automatically sharded, as it might not be the desired behavior.

If you want data to be unique cluster-wide, you have to explicitly tell Gatling to shard the data, e.g.:

```
val feeder = csv("foo.csv").shard
```

Assuming a CSV file contains 1000 entries, and 3 Gatling nodes, the entries will be distributed the following way:

- First node will access the first 333 entries
- Second node will access the next 333 entries
- Third node will access the last 334 entries



`shard` is available in Gatling OSS DSL but is a noop there. It's only effective when running tests with FrontLine.

## 1.5. Resolving Injector Location in Simulation

When running a distributed test from multiple locations, you could be interested in knowing where a given injector is deployed in order to trigger specific behaviors depending on location.

For example, you might want to hit `https://mydomain.co.uk baseUrl` if injector is deployed on AWS London, and `https://mydomain.com` otherwise.

You can resolve in your simulation code the name of the pool a given injector is deployed on:

```
val poolName = System.getProperty("gatling.frontline.poolName")
val baseUrl = if (poolName == "London") "https://domain.co.uk" else
  "https://domain.com"
```



This System property is only defined when deploying with FrontLine. It's not defined when running locally with any Gatling OSS launcher.

## 1.6. Publishing Fatjars into Binary Repositories

Instead of building tests from sources, you have the option of building binaries upstream and publishing them into a binary repository (JFrog Artifactory, Sonatype Nexus or AWS S3) so FrontLine just has to download them.



Please check your build tool documentation and the standards in your organization about the way to set credentials.

### 1.6.1. Maven

You'll have to configure either `repository` or `snapshotRepository` block whether you want to deploy releases or snapshots.

```
<distributionManagement>
  <repository>
    <id>your.releases.repository.id</id>
    <url>REPLACE_WITH_YOUR_RELEASES_REPOSITORY_URL</url>
  </repository>
  <snapshotRepository>
    <id>your.snapshots.repository.id</id>
    <url>REPLACE_WITH_YOUR_SNAPSHOTS_REPOSITORY_URL</url>
  </snapshotRepository>
</distributionManagement>
```

You'll need `frontline-maven-plugin` version 1.0.3 at least. Fatjar artifact will be automatically attached to your project and deployed with the `shaded` classifier.

```
mvn deploy
```

### 1.6.2. Gradle

The main idea is to use the official maven publish plugin and ask it to use the task named `frontLineJar`, then define a repository:

```
apply plugin: "maven-publish"

publishing {
    publications {
        mavenJava(MavenPublication) {
            artifact frontLineJar
        }
    }
    repositories {
        maven {
            if (project.version.endsWith("-SNAPSHOT")) {
                url "REPLACE_WITH_YOUR_SNAPSHOTS_REPOSITORY_URL"
            } else {
                url "REPLACE_WITH_YOUR_RELEASES_REPOSITORY_URL"
            }
        }
    }
}
```

You can deploy the test jar with the following command:

```
gradle publish
```

An artifact will be published will the `tests` classifier.

### 1.6.3. Sbt

```
packageBin in Test := (assembly in Test).value
publishArtifact in Test := true
publishTo :=
  (if (isSnapshot.value)
    Some("private repo" at "REPLACE_WITH_YOUR_SNAPSHOTS_REPOSITORY_URL")
  else
    Some("private repo" at "REPLACE_WITH_YOUR_RELEASES_REPOSITORY_URL")
)
```

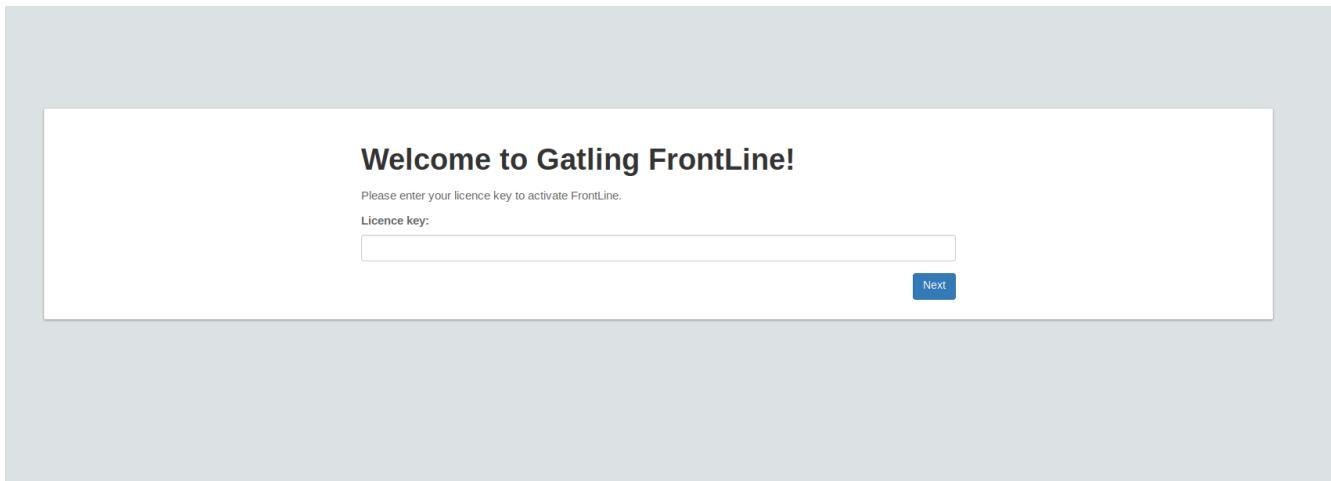
```
sbt test:publish
```

An artifact will be published will the `tests` classifier.

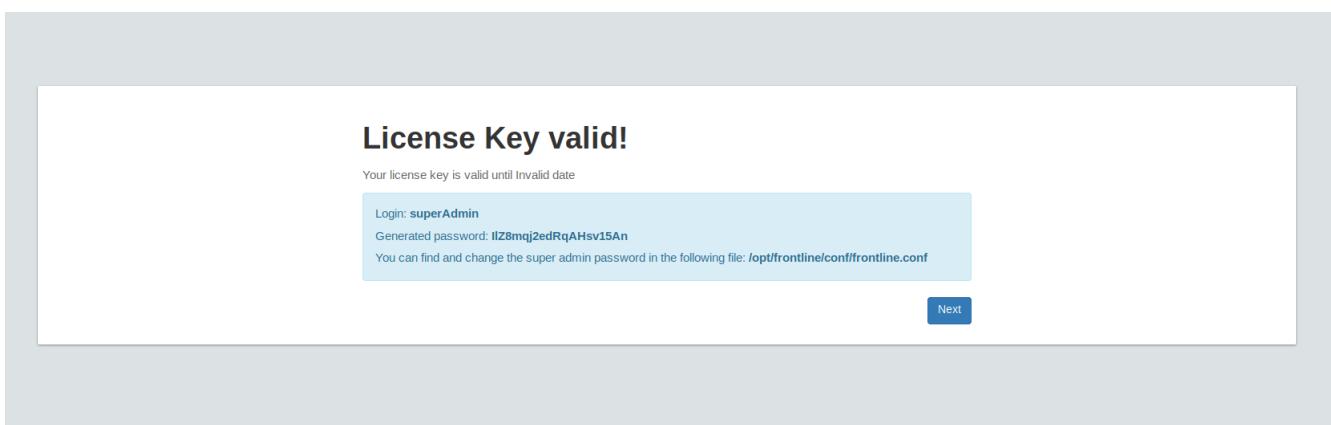
# Chapter 2. Dashboard User Guide

## 2.1. Configuration

The first step before using FrontLine is to configure your license key.



Once you've filled your license and clicked on the "Next" button you will get the credentials to connect to the superAdmin account. You can change this password in the `frontline.conf` file.



Click on the "Next" button to finish the configuration step and restart FrontLine.

## 2.2. Login

Logging in is necessary to use FrontLine. FrontLine is accessible by default on port 10542.



If it is the first time you use FrontLine, you can use the super admin account with the following credentials:

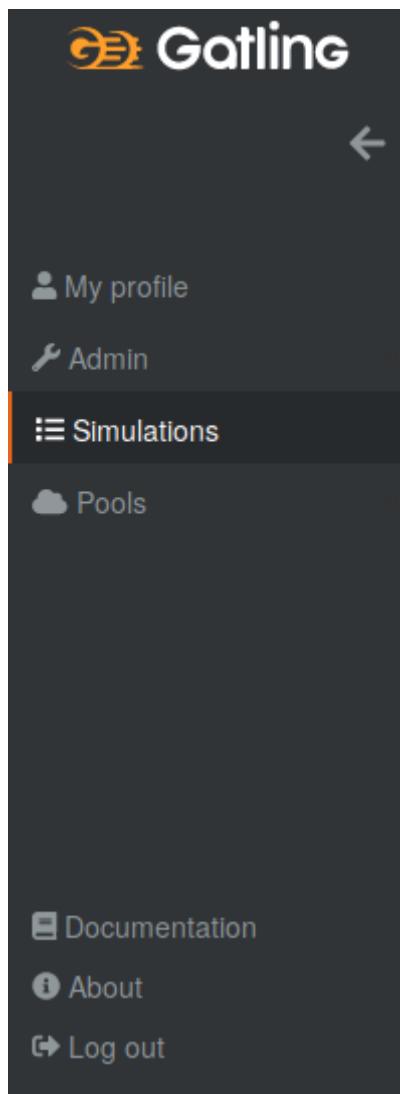
- Login: **superAdmin**
- Password: generated by the configuration step or in the `frontline.conf` file. You can change it in the `frontline.conf` file.

If you have configured the OpenID authentication, all of your user base can connect as a viewer. Make sure that FrontLine is allowed to open pop-ups on your browser, or you won't be able to login. To connect as a superAdmin, click on **Connect with basic credentials** and fill as previously explained. To modify your users permissions, you'll need to go the [users](#) page, and modify the users.



## 2.3. Overview

Once you are logged in, you are now able to navigate using the FrontLine navigation bar on the left side.



FrontLine is composed of:

- The Profile section, letting a user manage his account settings (not accessible to the superAdmin)
- The Admin section, letting administrators manage teams, users, API tokens, private keys and repositories
- The Simulations section, where runs, trends and simulations can be seen and managed

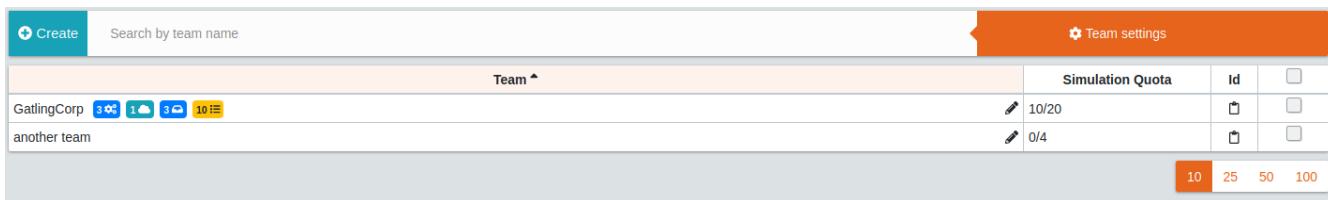
- The Pools section, where you can configure your cloud and on-premises instances
- The Documentation section, displaying all documentation guides, the link to the Swagger for our public API, and samples for your FrontLine tests
- The About section, letting you know the current version of FrontLine and the expiration date of your license key

We are now going to describe in depth each one of these sections.

## 2.4. Admin

### 2.4.1. Managing Teams

To access the Teams administration, click on **Admin** in the navigation bar, and choose **Teams**.



The screenshot shows a table titled 'Team' with two rows. The first row contains 'GatlingCorp' with icons for users (2), clouds (1), pools (3), and simulations (10). The second row contains 'another team'. To the right of the table are columns for 'Simulation Quota' (10/20 and 0/4), 'Id' (with copy icons), and a dropdown menu for page size (10, 25, 50, 100).

Team	Simulation Quota	Id	
GatlingCorp	10/20		
another team	0/4		

10 25 50 100

In the teams table, you can visualize the team name, the optional team quota, and the number of associated users, pools and simulations. You can also copy the team id by clicking on the icon.

#### Teams settings

To open teams settings, click on **Teams Settings** on the right side of the search bar.

### Edit Team Settings

**Simulation quotas:** **Associates a maximum number of simulations to each team**

**Initial quotas:**

Gatling Corp

20

Default

1

**Maximum simulations allowed:**

Enable simulations quotas if you need to restrict the number of simulations each team can create. When enabling this feature, you'll have to set a quota for each existing team.

#### Team

To create a team, click on the **Create** button.

## Create Team

Team name:	Great Team
Simulation quota:	20 89 slots remaining

You can edit the team by clicking on the icon and delete them using the checkboxes on the table's right part. Simulation quota only appear if simulations quotas are enabled.

The simulation quota of a team means the number of simulations a team is allowed to own. By default, there won't be any limitation, and your teams will be able to create simulations until you reach the number of simulations defined in your license.

To define a quota, please click on the **Team settings** button. The checkbox **Simulations quotas** needs to be enabled if you want the quotas to be applied. The sum of the quotas needs to be inferior to the number of simulations allowed by your license. Please note that if this option is enabled, you need to provide a quota for each team, or this team won't be able to create a simulation. Once this option is enabled, that quota may be edited in the team edition modal.

### Edit Team Settings

Simulation quotas:

Initial quotas:

another team	4
GatlingCorp	20

Maximum simulations allowed: 60

**Save** **Close**

### 2.4.2. Managing Users

To access the Users administration, click on **Admin** in the navigation bar, and choose **Users**.

#### Permissions

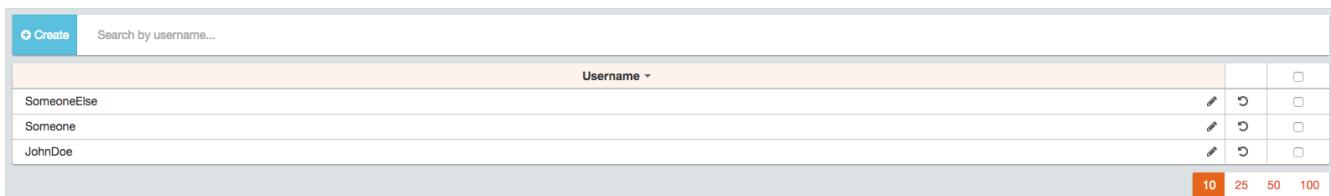
There are 4 different user roles in FrontLine:

- System Admin
- Team Admin
- Tester
- Viewer

	Viewer	Tester	Team Admin	System Admin
Access own profile 1	✓	✓	✓	✓
Access Reports and Trends	Own team	Own team	Own team	Own team
Start Simulation		Own team	Own team	Own team
Generate Public Links		Own team	Own team	Own team
Create Simulation			Own team	Own team
Access Pools			Own team	Own team
Manage Pools				Own team
Administratate Private Keys and Repositories			Own team	Own team
Administratate API Tokens, Users and Teams				Own team

Each role can be global or team-specific.

## User administration



The screenshot shows a user management interface with a search bar at the top. Below the search bar is a table with three rows, each representing a user: "SomeoneElse", "Someone", and "JohnDoe". Each row contains a checkbox in the first column and a small icon in the second column. At the bottom right of the table, there are buttons for page size selection: "10", "25", "50", and "100".



If you switch between FrontLine embedded user management system and LDAP/OpenID mode, or if there is a problem fetching your user data in LDAP/OpenID, some users may be flagged as invalid. For example, if a user created in FrontLine doesn't exist in your LDAP/OpenID server, you won't be able to connect with this user anymore.

To create a user, click on the **Create** button. Once the user is created, copy his password, as you won't be able to retrieve it again. OpenID authentication disable user creation, and only allow to edit users permissions (users who have already connected one time).

## Create User

**Username:** JohnDoe

**First Name:** John

**Last Name:** Doe

**Mail address:** john.doe@mail.com

**Global roles:**  System Admin  Test Admin  Tester  Viewer  None

**Specific roles:**

**Teams:**

**Save** **Close**

If you are using FrontLine with LDAP or OpenID, you will only have to fill the username and permissions to create a user, the other pieces of information come directly from the LDAP/OpenID. The username should be the same username as in the LDAP/OpenID.

You can edit the user by clicking on the icon and delete them using the checkboxes on the table's right part. To reset a user password, click on the icon (only available in non-LDAP mode).

It should be noted that the superAdmin account can't be deleted or even modified here.

### 2.4.3. Managing API Tokens

To access the API Tokens administration, click on **Admin** in the navigation bar, and choose **API token**.

Create	Search by token name...		
Name ^	Value		
New Token	16addca9...		
UsefulToken	4c366d0bf3c3d374d3a6908539daacf885ca126fb0ba176e2d9fb8957f9341dd		
10 25 50 100			

To create an API token, click on the **Create** button. Once the API token is created, copy the token, as you won't be able to retrieve it again.

# Create API Token

Api Token name:

Global Permissions:  Read  Start  All  None

Teams:

There are three permissions available for an API Token:

- The Start permission, allowing to start simulations (typically useful in a CI like Jenkins)
- The Read permission, allowing to read all the data from runs (typically useful in Grafana)
- The All permission, combining both of the previous permissions

You can edit the API Token permissions by clicking on the  icon on the right part of the table. A permission can be set globally or to a specific team. To regenerate a token, click on the  icon.

## 2.4.4. Managing Private Keys

To access the Private Keys administration, click on **Admin** in the navigation bar, and choose **Private Keys**. A private key corresponds to the ssh key used to connect to your Git repository or pool instances.

 Create	Search by private key or team name			
Team 	Name 	Scope 	Type 	Filename
Global	abc_rsa 	Pool	Path	abc_rsa
Global	aws-frankfurt	Pool	Path	id_frankfurt
Global	aws_irland	Repository	Path	id_irland
Global	gcp-key 	Pool	Path	gcp-key
Global	gitlab-cedric 	Repository	Uploaded	gitlab-cedric
Global	id_dont_use_reserved_for_ci 	Pool	Uploaded	id_dont_use_reserved_for_ci

To create a Private Key, click on the **Create** button.

## Create Private Key

**Name:** My AWS private key

**Team:** Global

**Scope:** repository pool all

**Private Key:** Upload Locate

id\_california ✖

**Save** **Close**

The screenshot shows a modal dialog titled "Create Private Key". It has several input fields: "Name" set to "My AWS private key", "Team" set to "Global", "Scope" with "pool" selected (other options are "repository" and "all"), and "Private Key" with "Upload" and "Locate" buttons. Below the fields is a list item "id\_california" with a red "X" icon. At the bottom right are "Save" and "Close" buttons.

A private key can be scoped on pool or repository. It means that you can only use this private key while configuring a repository or a pool. The all scope can't be chosen, as it is only there for the legacy private keys without scope.

You have two possibilities to reference private keys:

- Upload them directly by drag-and-drop or click on the input to choose the file on your filesystem
- Locate a private key existing on FrontLine's host. The private key permissions should be 600 or 400, and its owner should be the FrontLine process user



If you are using the AWS marketplace offer and wish to reference an existing private key, you must connect with the `ec2-user` user and then `sudo` to the `frontline` user which is the one running the FrontLine process.

You can edit the private key by clicking on the icon and delete them using the checkboxes on the table's right part.

### 2.4.5. Managing Repositories

To access the repositories administration, click on **Admin** in the navigation bar, and choose **Repositories**

There are 2 types of repository: the ones where you download and compile the sources, and the others where you download an already-compiled project

Admin > Repositories				
<span style="background-color: #0072bc; color: white; padding: 2px 10px; border-radius: 5px;">Create</span> <span style="margin-left: 10px;">Search by name...</span>				
Team	Name	Type		
Global	awsS3-repository	AWS S3 Bucket	<span style="color: #0072bc;">Edit</span>	<span style="color: #ccc;">Delete</span>
Global	binary-repository	Binary Repository - JFrog Artifactory	<span style="color: #0072bc;">Edit</span>	<span style="color: #ccc;">Delete</span>
Global	sources-repository	Git Repository	<span style="color: #0072bc;">Edit</span>	<span style="color: #ccc;">Delete</span>

10
25
50
100

To create a repository, click on the **Create** button. You can edit the repository by clicking on the icon and delete them using the checkboxes on the table's right part.

## General

### Create Repository

#### Name



#### Team

#### Repository type



- **Name:** the name that will appear on the repositories table.
- **Team:** set if the repository is global or owned by a team
- **Type:** the desired type of your repository

## Sources Repository

Choose **Build from sources** as repository type if you wish that FrontLine fetch and compile the sources of your Gatling simulation. In this page, you'll configure how to fetch the sources.

# Create Repository

## Name

## Team

 Global 

## Repository type

 Build from sources  Download from a binary repository

## Build type

## Method

 Clone with SSH  Clone with HTTPS

## Git Repository URL

URL to the Git Repository hosting your simulation class

## Git Branch or Tag

## Git SSH key

SSH key to connect to your Git repository

There are 3 different ways to retrieve your sources:

- **Clone a Git repository:** If you want to clone a git repository. You'll need to fill in the URL of the targeted repository, and the targeted git branch or tag (which can be overridden in the simulation configuration). If you're using ssh authentication, you can also fill in a previously added [private key](#) scoped on repository with **Git SSH key**. If you're using HTTPS authentication, you can setup an username and password.

- **Use a project on FrontLine's filesystem:** Use a project located on FrontLine's filesystem, fill in the path to project repository.
- **Check out from Source Code Control System:** Useful if you're using a code control system other than Git, or if you need a really specific Git command.

## Binary Repository

Choose **Download from a binary repository** if you already compiled your project and pushed to a binary repository.

## Create Repository

### Name

### Team

### Repository type

### Repository manager type

### Repository URL

[https://\[host\]/\[repository\]](https://[host]/[repository])

### Authentication settings - Optional

API Key for the specified user

We currently support 4 different providers: JFrog Artifactory, Sonatype Nexus 2 & 3, AWS S3.

If you're using an Artifactory or Nexus repository, you'll need to fill in the following fields:

- **Repository URL:** the URL of the targeted repository

- **Authentication settings:** the key for the jar you want to download
  - **Username:** the username of the user with sufficient permissions
  - **API Key:** [API key for the current user](#)



make sure to follow the Repository URL pattern as follows

- JFrog Artifactory : http[s]://<host>/<repository>
- Sonatype Nexus 2 : http[s]://<host>/nexus/content/repositories/<repository>
- Sonatype Nexus 3 : http[s]://<host>/repository/<repository>

How to find the repository URL?

- Artifactory :

The screenshot shows the JFrog Artifactory interface. On the left, there's a sidebar with various icons for navigation. The main area is titled "Artifact Repository Browser" and shows the "gatling" repository. The "General" tab is selected, displaying a message: "This repository is blacked out, items can only be viewed but cannot be resolved or deployed." Below this, there are fields for "Name" (gatling) and "Package Type" (Maven). A tooltip "Copy full file path to clipboard" points to the "Name" field. On the right, there's a "Actions" dropdown menu.

- Nexus :

The screenshot shows the Nexus Repository Manager interface. At the top, it says "The Apache Software Foundation" and "Log In". The main area is titled "Repositories" and shows the "Maven Staging Group" repository. The table lists various repositories like "Maven Staging Group", "Public Repository", "Snapshots", etc. On the left, there's a sidebar with "Artifact Search", "Views/Repositories", and "Help". On the right, there are sorting and filtering options, including "Sort Ascending", "Sort Descending", and "Columns".

If you're using an AWS S3 bucket, you'll need to fill in the following fields:

## Create Repository

### Name

### Team

### Repository type

### Repository manager type

### Profile name

*Profile described in the AWS credentials file or environment/system variables.*

### Bucket name

- **Profile name:** choose a profile described in `~/.aws/credentials`, or using permissions granted to the EC2 instance FrontLine is deployed on
- **Region:** the region where you created your bucket
- **Bucket name:** the bucket name

Before saving, we advice to check the connection to the repository by clicking on the **Check Connection** button.

## 2.5. Accessing your profile

You can view your profile at any time by clicking on the **My Profile** button in the navigation bar.

Team	Role
All Team	Tester
GatlingCorp	System Admin

The informations displayed are the following:

- Username
- First and Last name
- Mail address
- Different roles

To update your profile, modify the fields you want to change, then click on the button **Update your informations.**

You can't update your username here. You need to ask a System Admin to change your roles on the **Users Admin** page.

It should be noted that the superAdmin account doesn't have a profile.

If you are using FrontLine with LDAP, profiles will be read-only since users information are retrieved from the LDAP and not managed by FrontLine itself.

## 2.6. Pools

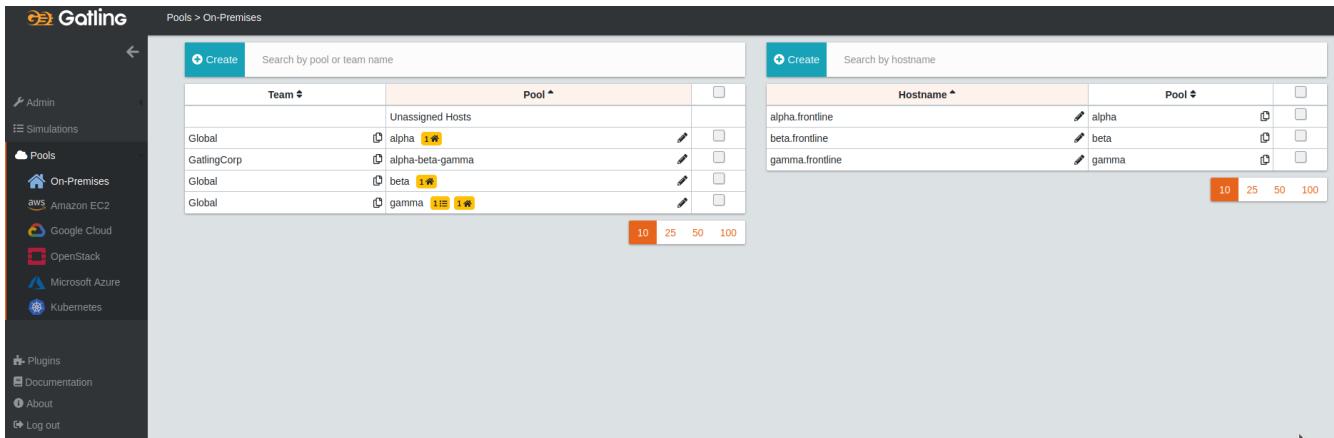
The Pools view is split by pool type, each in a different tab. There are currently 6 types of pool:

- **On-premises**: a pool which contains hosts configured manually
- **AWS**: used to spawn instances on a configured AWS account
- **GCE**: used to spawn instances on a configured GCE account
- **OpenStack**: used to spawn instances on a configured OpenStack account
- **Microsoft Azure**: used to spawn instances on a configured Microsoft Azure account
- **Kubernetes**: used to spawn instances through your configured Kubernetes/OpenShift

If you are using the AWS marketplace offer, you will only have access to the On-premises, AWS, and Kubernetes pools. If you are using the Azure marketplace offer, you will only have access to the On-premises, Azure, and Kubernetes pools.

## 2.6.1. Warning about Private Keys

Every pool except Kubernetes will need to have a [Private key](#) configured, and scoped on Pools. You won't be able to create one of these pools if you have not created a private key.



The screenshot shows the Gatling interface with the 'On-Premises' pool selected. The left sidebar includes options like Admin, Simulations, Pools, On-Premises (aws, Google Cloud, OpenStack), Microsoft Azure, and Kubernetes. The main area displays two tables: 'Pools > On-Premises' and 'Hosts > On-Premises'. The first table lists pools: 'Unassigned Hosts', 'alpha' (with 1 simulation), 'alpha-beta-gamma', 'beta' (with 1 simulation), and 'gamma'. The second table lists hosts: 'alpha.frontline' (with 1 simulation), 'beta.frontline' (with 1 simulation), and 'gamma.frontline' (with 1 simulation). Both tables have columns for 'Hostname' and 'Pool'.

## 2.6.2. On-premises

Here you can create an On-premises pool which represents existing machines.

The pool is defined by a name and a team:

### Create On-Premises Pool



The form for creating an On-Premises pool has fields for 'Pool name:' (containing 'New Pool') and 'Team:' (containing 'Global'). At the bottom right are 'Save' and 'Close' buttons.

You can edit the name and set if the pool is global or owned by a team by clicking on the  icon of that pool in the table. If you click on the name, it will highlight all the hosts which are in this pool in the host table.

You can delete one or more pool by selecting them and click on the **Delete** button above the table. The number of simulations associated to the pool is also displayed.

Now you need to create an host and assign it to the pool. To do so, you need to fill the following form:

# Create On-Premises Host

<b>Hostname:</b>	beta.frontline
<b>User:</b>	gatling
<b>Private Key:</b>	~/.ssh/abc_rsa
<b>Pool:</b>	alpha
<b>Working Directory:</b>	/tmp

Check Connection Save Close

An host is basically a server where the FrontLine server will upload the binary to use it as an injector for the test.

- **Hostname:** the hostname of the server
- **User:** the username to use for the connection to the server
- **Private Key:** the private key used by the server for SSH authentication
- **Pool:** Choose the pool to assign the host. You can select **Unassigned Hosts** if you don't want to put your host into a pool yet.
- **Working Directory:** Directory on the host where the code will be launched, it needs to be executable

To be sure that FrontLine can access to the host you can click on the **Check connection** button.

You can edit your hosts by clicking on the icon in the hosts table. You can duplicate an host to quickly create an host with the same parameters by clicking on the icon of the host on the table.

Hostname ▲	Pool ▲	
beta.frontline	alpha	

If you select one or more hosts, you will see a new button named **Action** appear on the search bar above the table which allow you to delete all the selected hosts or to switch pool to another for all selected hosts.

The screenshot shows a user interface for managing hosts and pools. At the top left is a blue button labeled "Create" with a plus sign icon. To its right is a search bar with the placeholder "Search by hostname...". On the far right is an orange "Action" button with a dropdown arrow. Below this is a table with two columns: "Hostname" and "Pool". The "Hostname" column has a header with an upward arrow icon. The "Pool" column has a header with a double-headed arrow icon. A single row is visible, showing "beta.frontline" in the Hostname column and "alpha" in the Pool column. To the right of this row is a white card with two options: "Switch Pool" (with a double-headed arrow icon) and "Delete" (with a trash bin icon). A small minus sign is located below the "Delete" option.

Hostname	Pool
beta.frontline	alpha

You can sort the pool table or the host table by clicking on their respective columns.

### 2.6.3. AWS Pool

An AWS Pool is a reference to the AWS account you want to use to spawn injectors to run the simulation. To configure the type of instances you want to spawn, you need to fill the form below:

## Create AWS Pool

Pool name:

Europe - Paris

Team:

Global

### Credentials Settings

Profile Name:

frontline

### Instance Settings

Region:

EU (Paris)

AMI:

Certified

Amazon Linux w/ OpenJDK 8

VPC

vpc-392d9150

Subnet

subnet-c1d666a8

Security Groups

HttpsRestricted (sg-0d16f0e95a2287450)  
HttpsUnrestricted (sg-064728e9af6d87580)  
HttpUnrestricted (sg-0afbddd3682b86751)  
**OpenBar (sg-0c827e79731b07fd6)**  
PrometheusRestricted (sg-005513d27f918add0)

Instance Type

t2.large

Key Pair:

paris

User Name:

ec2-user

Private Key:

AWS

Use Elastic IP:

Use private IP:

IAM Instance Profile:

ProbeLauncher@EC2

AWS Tags:

key

value



Save

Close

- **Team:** Set if the pool is global or owned by a team
- **Profile Name:** Name of the AWS profile described in the AWS credentials file. If you want to use System or Environment properties instead of this file, choose [Use environment or system variables](#)
- **Region:** the region where to spawn your instances
- **AMI:** the AMI you want to use for your instances. You can use our certified AMIs or the ID of your custom AMI (the AMI should at least have JDK8 installed, a configured key pair without password and the port 22 & 9999 should be open)
- **VPC:** the VPC in which your instance will be created
- **Subnet:** the subnet in which your instance will be created
- **Security Group:** the security groups the instance will use
- **Instance Type:** the type of the instances you want to spawn
- **Key Pair:** the Key pair name used by your AMI
- **User Name:** the username used by your ssh command to connect to the instances. If you use one of our certified AMIs, the username will be ec2-user
- **Private Key:** the previously added [private key](#) used by your AMI
- **Use Elastic IP:** Allow instances to use predefined Elastic IP
- **Use private IP:** FrontLine will use the injector private IP instead of the public one by default. If unchecked, the private IP remains a fallback if a public IP is missing. This option should be used only when the FrontLine host and the injector are both on AWS on the same network.
- **IAM Instance Profile:** optional step, you can specify an IAM instance profile to grant injectors permissions
- **AWS tags:** optional step, the tags will be visible in your AWS interface, hence you will be able to monitor them



If you're using our certified AMIs, make sure that you add a security group allowing Internet access. This is required for automatic critical security updates checks done by the OS.

#### 2.6.4. GCE Pool (On-premises license only)

Like the AWS Pool, a GCE Pool is a reference to the GCE account you want to use to spawn injectors to run the simulation. To configure the type of instances you want to spawn, you need to fill the form below:

## Create GCE Pool

Name: GCE Pool

Team: Global

### Credentials Settings

Credentials type: Application Default JSON

JSON File: ~/Documents/work/gatling/keys/gce/frontline-testing.json

### Instance Settings

Zone: europe-west1-b

Deployment: Image Instance Template

Image: Certified

OpenJDK 8

Machine type: n1-highcpu-4

Subnetwork: default

Preemptible:

Public SSH key:

ssh-rsa

Network tags:

ssh,http

Private key:

gcp-key

- **Team:** Set if the pool is global or owned by a team
- **Credentials:** If you're running Frontline on GCE or using `GOOGLE_APPLICATION_CREDENTIALS` to configure access, use *Application Default*. Otherwise, use JSON credentials.
- **Zone:** the zone where you want to spawn your injectors
- **Private Key:** the previously added [private key](#) used by your Template
- **Use private IP:** check this if you want your injectors to use their private IP instead of the public one
- **Deployment:** You can choose to spawn GCE instances from an image or an instance template

Specific configuration if you chose Image:

- **Image:** the image you want to use for your instances. You can use our certified Images or the url of your custom Image (the Image should at least have JDK8 installed and a configured key pair without password)
- **Machine type:** this machine type will be used by the injectors. We recommend using n1-highcpu-4 or n1-highcpu-8 machines.
- **Subnetwork:** the subnetwork the instances will use
- **Preemptible:** check this if you want to use preemptible instances (cheaper, but can be reclaimed by GCE)
- **Public SSH key:** content of the public SSH key used to connect to the instances
- **Network tags:** networks tags you may want to apply to the instances
- **Use Static IPs:** check this if you want your injectors to use predefined static IPs

Specific configuration if you chose Instance Template:

- **Template:** the template used for your instances, the template should at least have JDK8 installed, a configured key pair without password and the port 22 & 9999 should be open
- **Username:** the username used by your ssh command to connect to the instances

## 2.6.5. OpenStack Pool (On-premises license only)

Like the others pools, an OpenStack Pool is a reference to the OpenStack account you want to use to spawn injectors to run the simulation. We only support Keystone version 3. To configure the type of instances you want to spawn, you need to fill the form below:

**Name:** pool ovh

**Team:** Global

---

### Credentials Settings

---

**Identity Endpoint:** https://auth.cloud.ovh.net/v3

**Identity:** Default:\*\*\*\*\*

**Credentials:** \*\*\*\*\*

**Scope (Optional):** project:\*\*\*\*\*

---

### Instance Settings

---

**Region:** SBG1

**Key Pair:** frontline

**Network:** Ext-Net

**Image:** Debian 10

**Flavor:** b2-30-flex (vcpus: 8 - RAM: 30000Mo)

**Security Group:** OpenBar

**User Name:** admin

**Private Key:** id\_dont\_use\_reserved\_for\_ci

**Availability Zone:** nova

- **Team:** Set if the pool is global or owned by a team
- **Identity Endpoint:** complete url of your OpenStack API (eg: <https://auth.cloud.ovh.net/v3>)
- **Identity:** your OpenStack identity, it is most of the time: `domain:username`
- **Credentials:** Account password
- **Scope:** you can use scoped authentication (eg: `project:myprojectname`)
- **Region:** the region where to spawn your instances
- **Key Pair:** the Key pair name use by your image
- **Image:** the image used for your instances (the image should at least have jdk8 installed and a configured key pair without password and the port 22 & 9999 should be open)
- **Flavor:** the flavor used for your instances
- **Security Group:** the security group of your account
- **Username:** the username used by your ssh command to connect to the instances
- **Private Key:** the previously added [private key](#) used by your images
- **Availability Zone:** the optional availability zone

## 2.6.6. Microsoft Azure Pool (On-premises license and Azure Marketplace only)

A Microsoft Azure Pool is a reference to the Azure account you can use to spawn injectors to run the simulation. Only Linux virtual machines are supported.

To configure the type of instances you want to spawn, you need to fill the form below:

## Create Azure Pool

**Pool name:**

Name of your Azure pool

**Team:**

Global

### Credentials Settings

**Application Token**

**Managed Service Identity**

**Subscription ID:**

xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

**Tenant ID:**

xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

**Client ID:**

xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

**Client Secret:**

xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

### Instance Settings

**Region:**

westeurope

**Size:**

Standard\_F2 (2 cores - 4096Mb)

**Network:**

Your custom network

**Subnet:**

default

**Use private IP:**



**Image:**

Certified

JDK 8

**Public ssh key:**

ssh-rsa AAAAAaaaA.....

**User Name:**

azure-user

**Private Key:**

Custom Private Key

**Save**

**Close**

- **Team:** Set if the pool is global or owned by a team
- **Subscription ID:** the Azure subscription you want to use, check <https://www.inkoop.io/blog/how-to-get-azure-api-credentials/> to retrieve it
- **Tenant ID:** the Azure tenant you want to use
- **Client ID:** the id of the Azure client you want to authenticate with
- **Client Secret:** the key used to authenticate
- **Region:** the region where you want to spawn your instances
- **Size:** the size of the instances
- **Network:** the network configured on your Microsoft Azure account you want to use
- **Subnet:** the subnet you want to use
- **Image or Image URL:** the certified image or the url of the image you want to use for your instances. You can use our certified images or the url of your custom VHD: the image should at least have JDK8 installed, a configured key pair without password and the port 22 & 9999 should be open, see the [Azure documentation](#) if you want to make your own image
- **Public Key:** the public ssh key to connect to your instances
- **Username:** the username used by your ssh command to connect to the instances
- **Private Key:** the previously added [private key](#) associated with the public ssh key

It's also possible to use User Assigned Managed Identities, refer to the installation guide if you want to create a Managed Identity:

## Create Azure Pool

**Pool name:** Name of your Azure pool

**Team:** Global

**Credentials Settings**

Application Token Managed Service Identity

You can use [User Assigned Managed Identities](#) when running FrontLine from an Azure virtual machine. Make sure to create one before using this pool and assign it to this virtual machine.

Please refer to the [installation guide](#).

### 2.6.7. Kubernetes/OpenShift Pool

A Kubernetes/OpenShift Pool is a reference to your Kubernetes infrastructure.

To configure the type of instances you want to spawn, you need to fill the form below:

## Edit Kubernetes Pool

Name:

kubernetes-cluster

Team:

Global

### Credentials Settings

?

 Kubernetes URL:

https://kubernetes.gatling.io

?

 Service Account Token:

eyJhbGciOiJSUzI1NilsImtpZCI6IihVR3U0bVg2b2pOWIFIYml0ZVVy  
UzNkMUJpYjJyTEVHXzdYd2tXdzViTDQifQ.eyJpc3MiOiJrdWJlcm5l  
/

?

 Namespace:

frontline

### Instance Settings

Connection:

NodePort Ingress Route

?

 Root URL:

https://kubernetes.gatling.io/frontline/injectors

?

 TLS Secret:

frontline-injectors

Docker image:

Certified

JDK 11

?

 CPU request:

4

?

 CPU limit:

4

?

 Memory request:

1.5Gi

?

 Memory limit:

1.5Gi

Check Connection

Save

Close

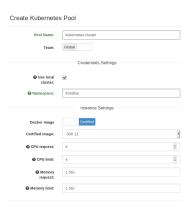
- Team: Set if the pool is global or owned by a team

- **Kubernetes Url:** The url of your Kubernetes API with the protocol
- **Service Account Token:** The token of your service account which has edit permissions on the namespace below (see [needed permissions](#))
- **Namespace:** The namespace/project name in which injectors will be spawned
- **Connection:**
  - **NodePort:** exposes the Service on each injector Node's IP at a static port
  - **Ingress:** exposes HTTP and HTTPS routes from outside the cluster to injectors within the cluster
    - **TLS secret name:** the optional secret containing a certificate used by the ingress ([TLS secrets documentation](#))
  - **Route:** (OpenShift extension) exposes HTTP routes (HTTPS not supported) from outside the cluster to injectors within the cluster.
    - **Secured:** allow you to add the desired certificate on the route ([OpenShift secured routes documentation](#))
      - **Certificate:** Certificate associated with the route
      - **Certificate key:** Certificate key associated to certificate
      - **CA Certificate:** Certificate authority signing the certificate
- **Docker Image:** Docker image that will be used for injectors. You can use our certified Docker images if your Kubernetes cluster has access to Docker Hub, or build your own with <https://github.com/gatling/frontline-injector-docker-image>
- **CPU request:** The minimum number of cores that you need for each one of your injector, express as cpus
- **CPU limit:** The limit of cores that you don't want your injector pod to exceed, express as cpus
- **Memory request:** The minimum memory that you need for each one of your injector
- **Memory limit:** The maximum memory that you need for each one of your injector

Limits and requests for memory are measured in bytes. You can express memory as a plain integer or as a fixed-point integer using one of these suffixes: E, P, T, G, M, K. You can also use the power-of-two equivalents: Ei, Pi, Ti, Gi, Mi, Ki.



If your FrontLine instance belongs to a Kubernetes cluster, you don't have to provide a **Kubernetes Url** and a **Service Account Token**. You can still choose to configure it, for example to create a pool in another cluster, by unticking **Use local cluster**. Also, you can specify preferring **Internal IP** over the **External IP** for connecting to Kubernetes nodes, by ticking **Prefer internal IP**.



## Minimal permissions for FrontLine service account

Service account associated to the [token](#) must be binded with permissions to manage services, nodes, routes, ingresses and pods (depending on your needs). Below, you can find a commented configuration file containing all needed permissions.

```
# Dedicated namespace for FrontLine
apiVersion: v1
kind: Namespace
metadata:
  name: frontline
---
# Service account named frontline
apiVersion: v1
kind: ServiceAccount
metadata:
  name: frontline-sa
  namespace: frontline
---
# Service account token
apiVersion: v1
kind: Secret
metadata:
  name: frontline-sa-token
  namespace: frontline
  annotations:
    kubernetes.io/service-account.name: frontline-sa
type: kubernetes.io/service-account-token
---
# Role containing needed permissions
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: frontline-manage-injectors
  namespace: frontline
rules:
  # Used to check the pool configuration
  - apiGroups: []
    resources: ["namespaces"]
    verbs: ["get"]
    # Needed for management of injectors instances
  - apiGroups: []
    resources: ["services", "pods", "pods/exec"]
    verbs: ["create", "delete", "get", "list", "patch", "update", "watch"]
    # Only for usage of Ingresses
  - apiGroups: ["extensions"]
    resources: ["ingresses"]
    verbs: ["create", "delete", "get", "list", "watch"]
    # Only for usage of OpenShift Routes
  - apiGroups: ["route.openshift.io"]
    resources: ["routes", "routes/custom-host"]
```

```

    verbs: ["create", "delete", "get", "list", "watch"]
---
# Bind role to the service account
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: frontline-role-binding
  namespace: frontline
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: frontline-manage-injectors
subjects:
  - kind: ServiceAccount
    name: frontline-sa
    namespace: frontline
---
# Only for usage of NodePort
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: frontline-manage-injectors
rules:
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["list"]
---
# Only for usage of NodePort
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: frontline-cluster-role-binding
subjects:
  - kind: ServiceAccount
    name: frontline-sa
    namespace: frontline
    apiGroup: ""
roleRef:
  kind: ClusterRole
  name: frontline-manage-injectors
  apiGroup: ""

```

## 2.7. Simulations

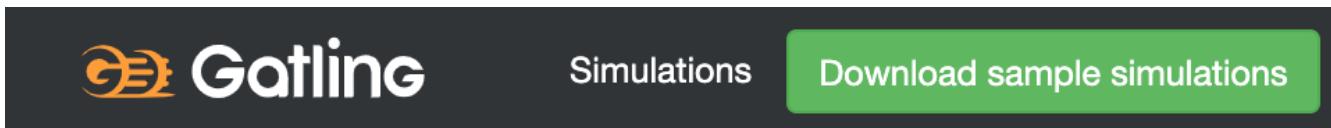
To access the Simulations section, click on **Simulations** in the navbar.

The Simulations view contains all the simulations you have configured and the result of their last run.

The screenshot shows the Gatling web interface with the 'Simulations' section selected. The main area is a table titled 'Simulation' with a header row. The columns are: Team, Name, Start, Runs, Build Start, Inject Start, Inject End, Duration, and Status. The 'Status' column uses color-coded icons: red for 'Assertions failed', pink for 'Broken', green for 'Successful', and grey for 'Build failed'. At the bottom of the table are navigation buttons (1, 2, 3, >>) and a page size selector (10, 25, 50, 100). A green button labeled 'Download sample simulations' is located at the top right of the interface.

Simulation										Last Run
Team	Name	Start	Runs	Build Start	Inject Start	Inject End	Duration	Status		
GatlingCorp	GatlingIoWorkload	⌚	▶	⌚	21/02/2019, 17:01:25	21/02/2019, 17:01:29	⌚	21/02/2019, 17:01:50	21s	Assertions failed
GatlingCorp	CapacityClosedWorkload	⌚	▶	⌚	12/09/2018, 18:01:17	⌚	12/09/2018, 18:01:45	⌚	1m 28s	Broken
GatlingCorp	AlphaFastWorkload	⌚	▶	⌚	11/09/2018, 16:31:16	11/09/2018, 16:31:42	⌚	11/09/2018, 16:33:14	1m 32s	Successful
GatlingCorp	IdleTimeoutSimulation	⌚	▶	⌚	25/07/2018, 13:48:19	25/07/2018, 13:48:43	⌚	25/07/2018, 13:49:24	41s	Successful
GatlingCorp	BasicSbtSimulation	⌚	▶	⌚	23/07/2018, 17:45:56	⌚	—	⌚	0s	Build failed
GatlingCorp	Http2Workload	⌚	▶	⌚	22/06/2018, 15:55:59	22/06/2018, 15:56:24	⌚	22/06/2018, 15:56:34	10s	Successful
GatlingCorp	OpenWorkload	⌚	▶	⌚	05/04/2018, 01:18:27	05/04/2018, 01:18:52	⌚	05/04/2018, 01:19:27	35s	Successful
GatlingCorp	ClosedWorkload	⌚	▶	⌚	05/04/2018, 01:17:05	05/04/2018, 01:17:29	⌚	05/04/2018, 01:18:03	34s	Successful
GatlingCorp	DelayWorkload	⌚	▶	⌚	07/03/2018, 12:02:37	07/03/2018, 12:03:01	⌚	07/03/2018, 12:03:39	38s	Successful
GatlingCorp	Cake	⌚	▶	⌚	22/02/2018, 15:22:42	⌚	—	⌚	0s	Broken

If you don't have any simulations configured yet and don't know how to start, you can download some FrontLine pre-configured projects by clicking on the "Download sample simulations" green button.



Those samples are ready to use maven, sbt and gradle projects with proper configuration for FrontLine. You can also download those samples with the download link in the Documentation section.

Back to the Simulations section, at the top, there is an action bar which allow several actions:

- Create a simulation
- Search by simulation or team name
- Edit global properties
- Delete selected simulations

The screenshot shows the 'Global Properties' section with a single entry: 'Fast workload' under 'Gatling team'. The table has columns for Team, Name, Start, Runs, Build Start, Inject Start, Inject End, Duration, and Status. The status is 'Successful'. A green button labeled 'Global Properties' is located at the top right.

Simulation										Last Run	
Team	Name	Start	Runs	Build Start	Inject Start	Inject End	Duration	Status			
Gatling team	Fast workload	⌚	▶	⌚	9/22/2020, 4:55:10 PM	⌚	9/22/2020, 4:55:22 PM	⌚	9/22/2020, 4:56:41 PM	1m 19s	Successful

## 2.7.1. Global Properties

Global properties contains every JVM options and system properties used by all of your simulations by default. Editing those properties will be propagated to all the simulations.

If you don't want to use the default properties, check [Use custom global properties](#) and enter your own.

# Edit Global Properties

Use custom Global Properties

## JVM Options:

```
-server  
-Xmx1G  
-XX:+UseG1GC  
-XX:+ParallelRefProcEnabled  
-XX:MaxInlineLevel=20  
-XX:MaxTrivialSize=12  
-XX:-UseBiasedLocking
```

## System Properties:

key

value



Save

Close

If you want specific properties for a simulation, you will be allowed to ignore those properties by checking the **Override Global Properties** box when creating or editing the simulation:



#### Global JVM Options - *Optional*

```
-server  
-Xmx1G  
-XX:+UseG1GC  
-XX:+ParallelRefProcEnabled  
-XX:MaxInlineLevel=20  
-XX:MaxTrivialSize=12
```

#### Simulation JVM Options - *Optional*

```
JVM options...
```

### 2.7.2. Creating a simulation



FrontLine has a hard run duration limit of 7 days and will abort any test running for longer than that. This limit exists for both performance (data who grow too humongous to be presented in the dashboard) and security (forgotten test running forever) reasons.

In order to create a simulation click on the "Create" button in the simulations table. There are 6 steps to create a simulation, 3 of which are optional.

#### Step 1: General

## Create Simulation: General



### Name

Great Simulation

### Team

Great Team

### Class name

Package and name of your simulation class

com.company.gatling.Simulation

Next >

Close

- **Name:** the name that will appear on the simulations table.
- **Team:** the team which owns the simulation.
- **Class name:** the package and the name of your simulation scala class in the project that you want to start.

## Step 2: Build configuration

In this step, you'll describe which [repository](#) FrontLine will use, and how to use it.

## Edit Simulation: Build configuration



### Build type

Build from sources

Advanced build configuration

### Repository

Company Repository

### Build command

The command to build your project

Maven project

◀ Previous

Next >

Save

Close

- **Build type:** How you want to retrieve and build your simulation. You may choose to build from sources, download a binary from a Sonatype Nexus or JFrog Artifactory repository, or download

a binary from an AWS S3 bucket.

- **Repository:** The [repository](#) you created previously

#### Option 1: Build from sources

In this step, FrontLine will download the sources from your repository, and compile them.

- **Build command:** the command to build your project. Three common commands are built-in for projects whose build tools configuration follow our installation guide:

- `mvn clean package -DskipTests --quiet` for maven project
- `sbt -J-Xss100M ;clean;test:assembly -batch --error` for sbt project
- `gradle clean frontLineJar -quiet` for gradle project



Please make sure that the tools you are using are installed and available on the FrontLine machine, for example: `mvn`, `sbt`, `git`, and `ssh`.

You can provide optional settings if you toggle **Advanced build configuration**.

- **Relative path:** the path to your simulation project in your repository (eg the Gatling simulation is not at the root of your git)
- **Environment variables:** the environment variables to be used along the build command. You can add as many environment variables as you want
- **Git Branch or Tag:** if you're using a git repository, you may specify another branch or tag than the one configured in the repository configuration

#### Option 2: Download binary from repository

In this step, you'll describe how FrontLine will download a jar deployed in a previously added repository. This jar must have been built with the same maven/sbt/gradle configuration as described in the Developer section in this guide.

## Edit Simulation: Build configuration



### Build type

Download binary from repository

### Repository

Gatling Repository (Sonatype Nexus 2)

### Artifact Maven coordinates

groupId

io.gatling

artifactId

demo

version

1.0

classifier - Optional

classifier

◀ Previous

Next ▶

Save

Close

- **Artifact Maven coordinates:** the maven coordinates of the desired artifact. *version markers are not supported*

### Option 3: Download binary from AWS S3

In this step, you'll describe how FrontLine will download a jar deployed in an AWS S3 bucket. This jar must have been built with the same maven/sbt/gradle configuration as described in the Developer section in this guide.

## Edit Simulation: Build configuration



### Build type

Download binary from AWS S3

### Repository

S3 Bucket

### Key

frontline-test-0.1.0-SNAPSHOT-shaded-demo

◀ Previous

Next ▶

Save

Close

- **Key:** the key for the jar you want to download

### Step 3: Pools configuration

In this step, you'll configure the pools used for the FrontLine injectors.

Edit Simulation: Pools configuration

GENERAL      BUILD      **POOLS**      JVM OPTIONS      SYSTEM PROPS      TIME WINDOW

**Pools**  
Pools used with the number of machines and the optional weight in %

Local (Demo Purpose Only)  1

**Weight distribution**  
Send an even number of requests from each pool, or a custom distribution

Even weights

< Previous    More options >    Save    Close

- **Weight distribution:** on even, every injector will produce the same load. On custom, you have to set the weight in % of each pool (eg the first pool does 20% of the requests, and the second does 80%). The sum of the weight should be 100%.
- **Pools:** defines the pools to be used when initiating the FrontLine injectors, see the section about [pools](#). You can add many pools with a different number of hosts to run your simulation. If you have more hosts than needed on your Pool, the hosts will be chosen randomly between all hosts available in this Pool.

After this step, you can save the simulation, or click on **More options** to access optional configuration.

### Step 4 & 5: JVM options & Java System Properties

These steps allows you to defines JVM arguments and system properties used when running this particular simulation. You can choose to override the global properties.

## Edit Simulation: JVM Options

1 GENERAL    2 BUILD    3 POOLS    4 JVM OPTIONS    5 SYSTEM PROPS    6 TIME WINDOW

Ignore Global Settings

**Global JVM Options - Optional**

```
-server  
-Xmx1G  
-XX:+UseG1GC  
-XX:MaxGCPauseMillis=30  
-XX:G1HeapRegionSize=16m  
-XX:InitiatingHeapOccupancyPercent=75
```

**Simulation JVM Options - Optional**

```
JVM options...
```

[Back](#) [Next](#) [Save](#) [Close](#)

## Edit Simulation: System properties

1 GENERAL    2 BUILD    3 POOLS    4 JVM OPTIONS    5 SYSTEM PROPS    6 TIME WINDOW

**Global System Properties**

java.net.preferIPv4Stack	true
java.net.preferIPv6Addresses	false

**Simulation System Properties - Optional**

key0	value0	-	
key1	value1	-	+

[◀ Previous](#) [Next ▶](#) [Save](#) [Close](#)

**i** JVM options and Java System Properties will be saved in a snapshot that will be available in the run. This information will be visible by anyone who has read access. You can exclude some properties from being copied if you prefix them with **sensitive..**

**i** You can configure the `gatling.frontline.groupedDomains` System property to group connection stats from multiple subdomains and avoid memory issues when hitting a very large number of subdomains. For example, setting this property as `.foo.com, .bar.com` will consolidate stats for `sub1.foo.com, sub2.foo.com, sub1.bar.com, sub2.bar.com` into `*****.foo.com` and `*****.bar.com`.

## Step 6: Time window

Configuring a ramp up or ramp down means that the start and end of your simulation won't be used for calculating metrics and assertions.

Edit Simulation: Time window

GENERAL      BUILD      POOLS      JVM OPTIONS      SYSTEM PROPS      TIME WINDOW

**Ramp up - Optional**  
The first seconds you want to exclude from the run  
 ▲ ▼

**Ramp down - Optional**  
The last seconds you want to exclude from the run  
 ▲ ▼

◀ Previous Save Close

- **Ramp Up:** the number of seconds you want to exclude at the beginning of the run.
- **Ramp Down:** the number of seconds you want to exclude at the end of the run.

**i** Ramps parameters will only be applied if the run duration is longer than the sum of the two.

### 2.7.3. Simulations table

Now that you have created a simulation, you can start it by clicking on the icon in the **Start** column of the table.

Simulation				Last Run				
Team	Name	Start	Runs	Build Start	Inject Start	Inject End	Duration	Status
Nice Team	SampleSimulation			11/16/2018, 2:49:59 PM	11/16/2018, 2:50:33 PM	11/16/2018, 2:50:56 PM	23s	<span style="background-color: #2e7131; color: white; border: 1px solid #2e7131; padding: 2px;">Successful</span>

A run have the following life cycle:

- **Building:** in which it will download and build the simulation from the sources, preparing the hosts if needed
- **Deploying:** in which it will deploy the simulation to run on all the hosts
- **Injecting:** in which the simulation is running and viewable from the Reports

Simulation				Last Run					⋮
Team ♦	Name ♦	Start	Runs	Build Start ▾	Inject Start ♦	Inject End ♦	Duration ♦	Status ♦	⋮
Nice Team	SampleSimulation	✎	⌚	⌚ 11/19/2018, 4:20:25 PM	⌚ 11/19/2018, 4:20:50 PM	⌚ 11/19/2018, 4:20:55 PM	4s	Injecting	✗ Abort

By clicking on the ⌚ icon in the **Build Start** column, Frontline will display the build logs of the simulation. There is a limit of 1000 logs for a run.

## Logs

```
[14:47:43,423] [warn] Merging 'META-INF/maven/io.netty/netty-transport-native-epoll/pom.properties' with strategy 'discard'
[14:47:43,423] [warn] Merging 'META-INF/maven/io.netty/netty-transport-native-epoll/pom.xml' with strategy 'discard'
[14:47:43,423] [warn] Merging 'META-INF/maven/io.netty/netty-transport/pom.properties' with strategy 'discard'
[14:47:43,423] [warn] Merging 'META-INF/maven/io.netty/netty-transport/pom.xml' with strategy 'discard'
[14:47:43,423] [warn] Merging 'META-INF/maven/org.apache.geronimo.specs/geronimo-jms_1.1_spec/pom.properties' with strategy 'discard'
[14:47:43,423] [warn] Merging 'META-INF/maven/org.apache.geronimo.specs/geronimo-jms_1.1_spec/pom.xml' with strategy 'discard'
[14:47:43,423] [warn] Merging 'META-INF/maven/org.asynchttpclient/async-http-client/pom.properties' with strategy 'discard'
[14:47:43,423] [warn] Merging 'META-INF/maven/org.asynchttpclient/async-http-client/pom.xml' with strategy 'discard'
[14:47:43,423] [warn] Merging 'META-INF/maven/org.asynchttpclient/netty-codec-dns/pom.properties' with strategy 'discard'
[14:47:43,423] [warn] Merging 'META-INF/maven/org.asynchttpclient/netty-codec-dns/pom.xml' with strategy 'discard'
[14:47:43,423] [warn] Merging 'META-INF/maven/org.asynchttpclient/netty-resolver-dns/pom.properties' with strategy 'discard'
[14:47:43,423] [warn] Merging 'META-INF/maven/org.asynchttpclient/netty-resolver-dns/pom.xml' with strategy 'discard'
[14:47:43,424] [warn] Merging 'META-INF/maven/org.javassist/javassist/pom.properties' with strategy 'discard'
[14:47:43,424] [warn] Merging 'META-INF/maven/org.javassist/javassist/pom.xml' with strategy 'discard'
[14:47:43,424] [warn] Merging 'META-INF/maven/org.jctools/jctools-core/pom.properties' with strategy 'discard'
[14:47:43,424] [warn] Merging 'META-INF/maven/org.jctools/jctools-core/pom.xml' with strategy 'discard'
[14:47:43,424] [warn] Merging 'META-INF/maven/org.slf4j/jul-to-slf4j/pom.properties' with strategy 'discard'
[14:47:43,424] [warn] Merging 'META-INF/maven/org.slf4j/jul-to-slf4j/pom.xml' with strategy 'discard'
[14:47:43,424] [warn] Merging 'META-INF/maven/org.slf4j/log4j-over-slf4j/pom.properties' with strategy 'discard'
[14:47:43,424] [warn] Merging 'META-INF/maven/org.slf4j/log4j-over-slf4j/pom.xml' with strategy 'discard'
[14:47:43,424] [warn] Merging 'META-INF/maven/org.slf4j/slf4j-api/pom.properties' with strategy 'discard'
[14:47:43,424] [warn] Merging 'META-INF/maven/org.slf4j/slf4j-api/pom.xml' with strategy 'discard'
[14:47:43,424] [warn] Merging 'META-INF/services/com.fasterxml.jackson.core.JsonFactory' with strategy 'filterDistinctLines'
[14:47:43,615] [warn] Strategy 'discard' was applied to 63 files
[14:47:43,615] [warn] Strategy 'filterDistinctLines' was applied to a file
[14:47:43,615] [warn] Strategy 'rename' was applied to 8 files
[14:47:43,749] [info] SHA-1: d84499bc41a0ffba441d0d8d69167681531866cd
[14:47:43,751] [info] Packaging /tmp/8731396520718514571/target/scala-2.11/root-8731396520718514571-test-0.1-SNAPSHOT.jar ...
[14:47:47,910] [info] Done packaging.
[14:47:47,924] [success] Total time: 15 s, completed 12 juil. 2016 14:47:47
[14:47:49,283] Simulation deployment completed successfully
```

Close

You can click on the 🔎 icon next to the status (if there is one) to display the assertions of the run. Assertions are the assumptions made at the beginning of the simulation to be verified at the end:

## Assertions

Global: max of response time is less than 1000.0

Succeeded with value 120

Close

## 2.7.4. Useful tips

- You can edit the simulation by clicking on the ✎ icon next to his name
- You can search a simulation by his name, or its team name
- You can sort the simulations by any column except the **Start** one
- A **Delete** button will appear on the action bar when you select a simulation, you will be able to delete all the selected simulations

- When a simulation is running, you can abort the run by clicking on the Abort button
- You can copy a simulation ID by clicking on the  icon next to his name

Be aware that deleting a simulation will delete all the associated runs.

## 2.7.5. Run / Trends

Runs list and trends can be accessed by clicking on the  icon in the simulations table.

This view contains the list of your simulation's runs which can be filtered by name and/or status and the Trends which are displaying information between those runs.

#	Run Snapshot	Build Start	Inject Start	End at	Duration	Status
85		21/02/2019, 17:01:25	21/02/2019, 17:01:29	21/02/2019, 17:01:50	21s	Assertions failed
84		21/02/2019, 16:59:53	—	—	0s	Deployment failed
83		21/02/2019, 16:57:09	—	—	0s	Deployment failed
82		21/02/2019, 16:54:33	—	—	0s	Deployment failed
81		21/02/2019, 16:52:56	—	—	0s	Deployment failed
80		21/02/2019, 16:50:47	—	—	0s	Deployment failed
79		20/02/2019, 01:45:49	20/02/2019, 01:46:04	20/02/2019, 01:46:23	19s	Assertions failed
78		20/02/2019, 01:29:09	20/02/2019, 01:29:23	20/02/2019, 01:30:33	1m 10s	Assertions failed
77		20/02/2019, 01:23:42	20/02/2019, 01:23:59	20/02/2019, 01:25:05	1m 6s	Assertions failed
76		20/02/2019, 01:16:29	20/02/2019, 01:16:45	20/02/2019, 01:17:45	1m	Assertions failed

1 2 3 ... > » 10 25 50 100

scenario: \* group: — request: \* limit: 10

**Requests and Responses**

**Response Time Percentiles**

**Throughput**

## Runs table

#	Run Snapshot	Build Start	Inject Start	End at	Duration	Status
264		3/29/2019, 4:49:56 PM	3/29/2019, 4:50:18 PM	3/29/2019, 4:50:38 PM	20s	✓ Assertions successful
263		3/21/2019, 11:34:34 AM	3/21/2019, 11:35:47 AM	3/21/2019, 11:36:07 AM	20s	Assertions failed
260		3/21/2019, 11:24:03 AM	3/21/2019, 11:25:20 AM	3/21/2019, 11:25:42 AM	22s	Assertions failed
234		3/20/2019, 11:07:11 AM	3/20/2019, 11:07:31 AM	3/20/2019, 11:07:51 AM	20s	✓ Assertions successful
233		3/20/2019, 11:04:28 AM	3/20/2019, 11:04:50 AM	3/20/2019, 11:05:10 AM	20s	✓ Assertions successful
232		3/15/2019, 11:29:34 AM	—	—	0s	Deployment failed
220		2/1/2019, 2:22:35 PM	2/1/2019, 2:22:56 PM	2/1/2019, 2:27:53 PM	4m 57s	Manually stopped
217		1/31/2019, 4:27:49 PM	1/31/2019, 4:28:04 PM	1/31/2019, 4:44:49 PM	16m 45s	Assertions failed
216		1/31/2019, 4:02:39 PM	1/31/2019, 4:02:56 PM	1/31/2019, 4:19:41 PM	16m 45s	Assertions failed
215		1/31/2019, 3:25:39 PM	1/31/2019, 3:25:54 PM	1/31/2019, 3:24:39 PM	16m 45s	Assertions failed

1 2 3 ... > » 10 25 50 100

Like the result of the latest run in the [simulations table](#) you have access to the [logs](#) of the run by clicking on the  icon and you can sort the table by each columns. The logs are only available for runs which are not flagged as "Successful".

If there is one, You can click on the  icon next to the status to display the [assertions](#) of the run. You can delete runs by selecting them and click on the **Delete** button in the action bar above the table.

You can comment a run by clicking on the  icon on the right side of the table.

Run Comments

---

Provide comments to the '#85-GatlingIoWorkload' run

2 injectors - new version of the script

Everything went pretty well

---

Save Close

You can also click on the  icon to see a snapshot of the run configuration. The system properties beginning with [sensitive](#). are not displayed.

# Run Configuration

## Simulation Class

frontline.sample.BasicSimulation

## Triggered by

API Token: test jenkins

## JVM Options

```
-server  
-Xmx1G  
-XX:+UseG1GC  
-XX:MaxGCPauseMillis=30  
-XX:G1HeapRegionSize=16m  
-XX:InitiatingHeapOccupancyPercent=75  
-XX:+ParallelRefProcEnabled  
-XX:+PerfDisableSharedMem  
-XX:+AggressiveOpts  
-XX:+OptimizeStringConcat  
-XX:+HeapDumpOnOutOfMemoryError
```

## System Properties

java.net.preferIPv4Stack

true

java.net.preferIPv6Addresses

false

## Pool Configuration

Pool Name	Pool Deployment	Number of Injectors	Instance Type	Region
Local	Local	1	N/A	N/A

Close

## Run Comparison

		Compare		Run 195 - 1/22/2019, 11:52:17 AM		&	Run 196 - 1/22/2019, 2:06:17 PM		
Request ^		Mean Response Time (ms)				Errors Ratio (%)			
		run 195 ↴	run 196 ↴	Delta ↴	Variance ↴	run 195 ↴	run 196 ↴	Delta ↴	Variance ↴
Global		9 231	9 223	-8	-0.09%	100	100	0	0%
/ group1 / group11 / group111 / jsonbc1k		9 276	9 299	+23	0.25%	100	100	0	0%
/ group1 / group11 / group112 / jsonbc1k		8 983	9 300	+317	3.53%	100	100	0	0%
/ group1 / group11 / jsonbc1k		9 560	9 247	-313	-3.27%	100	100	0	0%
/ group1 / group12 / jsonbc1k		8 464	8 109	-355	-4.19%	100	100	0	0%
/ group1 / jsonbc10k		9 152	9 646	+494	5.40%	100	100	0	0%
/ json		9 395	9 133	-262	-2.79%	100	100	0	0%
/ other group / jsona		9 415	9 678	+263	2.79%	100	100	0	0%
/ other group / jsona1k		9 586	9 382	-204	-2.13%	100	100	0	0%

OK

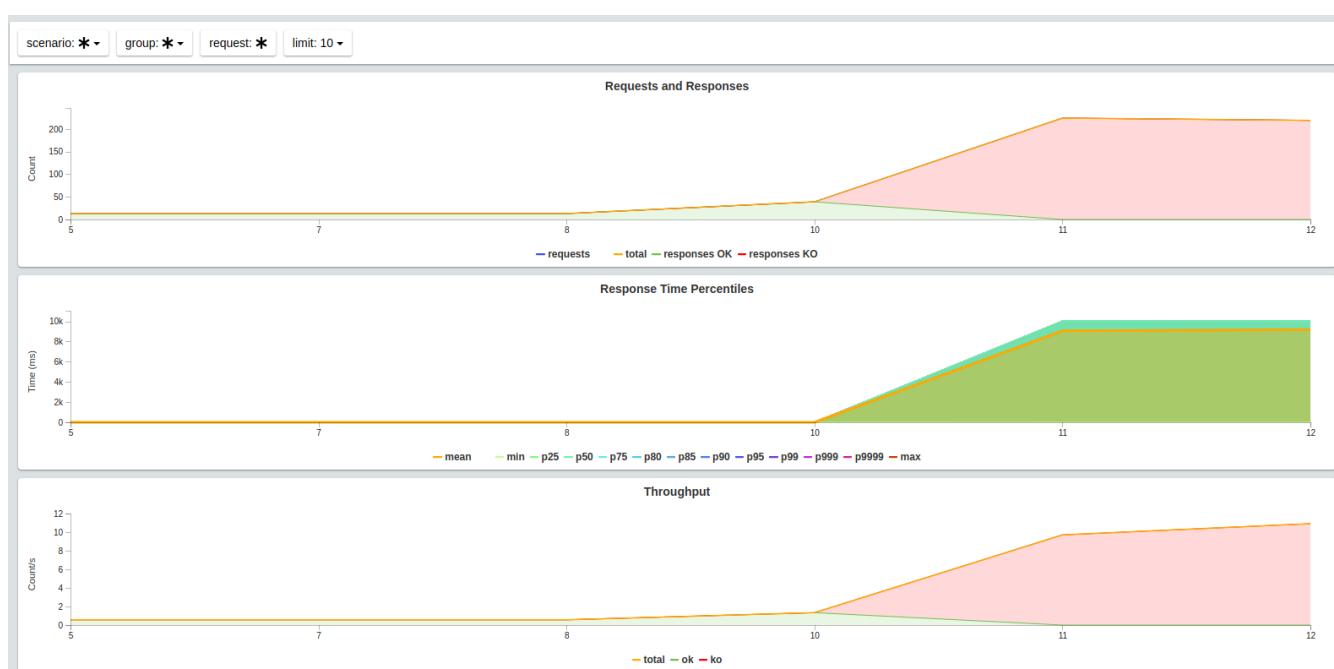
You can compare the results of two runs if you click on the "Compare runs" button in the table. It allows you to compare the response time and errors of the two runs for each request.

You can choose the specific metric you want to compare by clicking on the metric name, and the specific run you want to compare by clicking on the run number.

The delta and variance will be displayed, so you can check if there is a progression or a degradation in performance.

## Trends charts

The trends are charts that will display some global statistics for each run (eg: requests count) so that you can easily see how well your runs went compared to each other. Each run is represented by his number in the chart and the chart won't display the statistics of a failed run (eg: Timeout, broken, etc..).



You can filter the statistics shown by filtering through scenarios, groups or requests that are involved in each run. You can choose how many runs will be compared by changing the limit (10, 25, 50, 100):

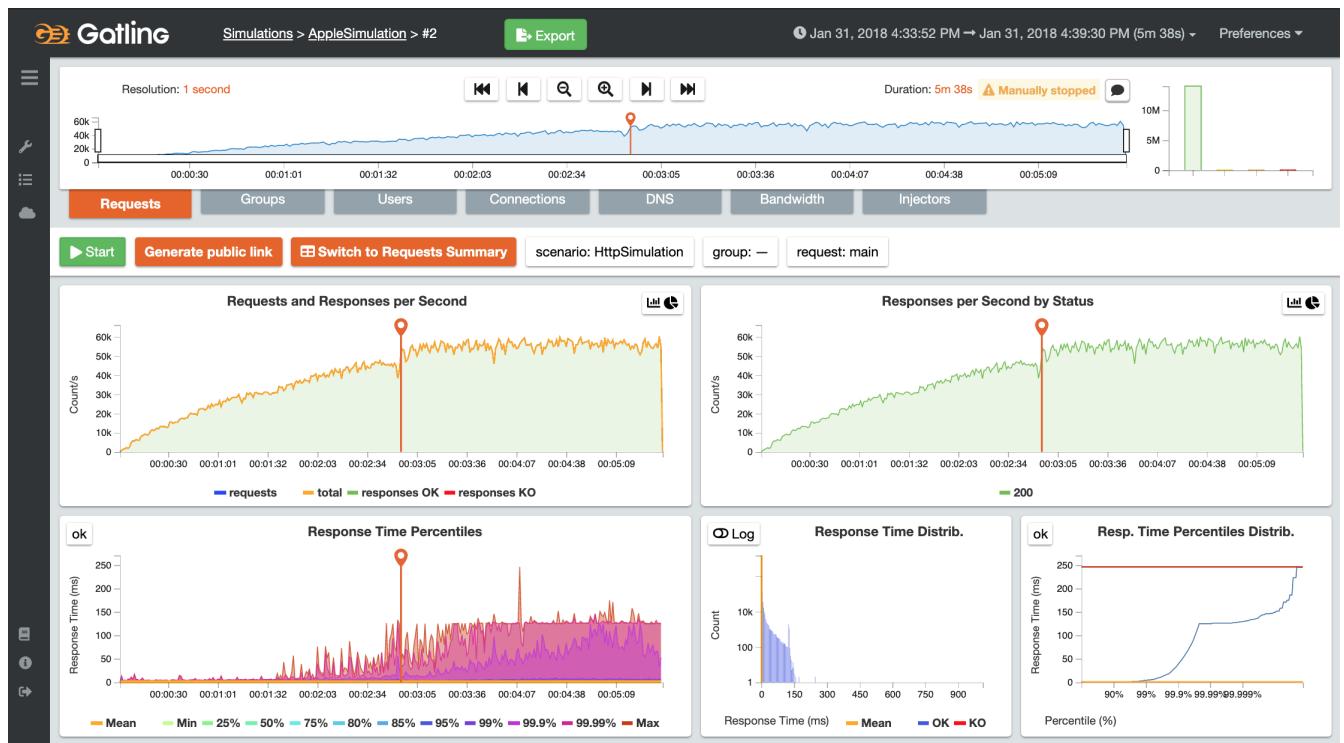
scenario: *	group: *	request: *	limit: 10 ▾
-------------	----------	------------	-------------

## 2.7.6. Reports

The reports can be accessed by clicking on the  icon in the [simulation table](#) or in the [runs table](#).

This view introduce all the metrics available for a specific run. This page consists of:

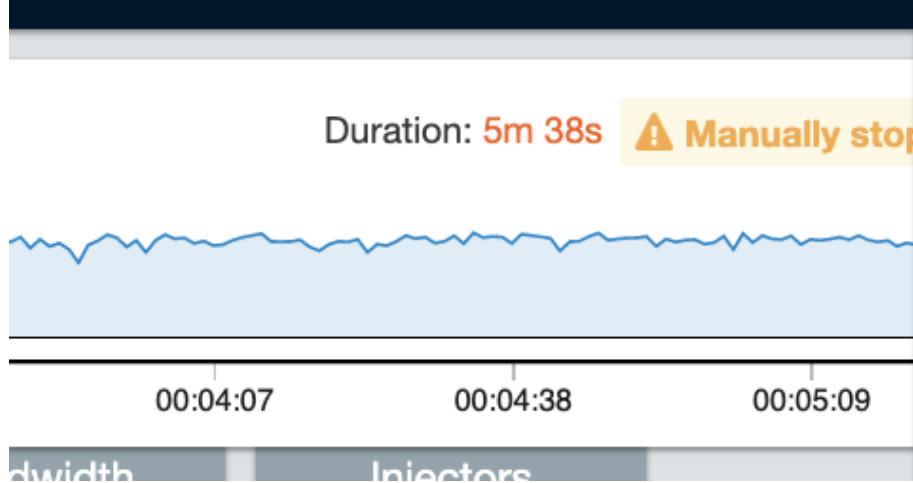
- [The top navigation bar](#)
- [The timeline](#)
- [Tabs](#)
- [The run bar](#)
- [Charts area](#)
- [The summary](#) (only for requests and groups tabs)
- [Export PDF](#)



### Top Navigation Bar

The navigation bar enable you to choose the simulation time range.

⌚ Jan 31, 2018 4:33:52 PM → Jan 31, 2018 4:39:30 PM (5m 38s) ▾



Last 5 minutes

Last 15 minutes

Last 30 minutes

Last 1 hour

Last 1 day

Everything

## Timeline

The timeline contains metrics of the full run providing an overview of the run. Global informations are available such as the resolution and the simulation name.

The resolution indicates the number of seconds per data point in the graph.

You can change the time range with control buttons or by selecting a region on the timeline:



## Assertions

The label below is used to display the status of the simulation (Ongoing, successful, timeout...). If your simulation has assertions, this label will be clickable to show the assertions results. You can comment the run run by clicking on the 🗣 icon.

Duration: 1m 32s ✓ Successful



## Tabs

Below the navigator chart, there are tabs to switch charts. Each tab has the same structure except the summary that is available only for requests and groups tabs.



## Run Bar

This bar is a combination of buttons:

- **Start / Abort:** Use this button to start a new run of the simulation, or stop the ongoing run (not available if you have a Viewer permission)
- **Grafana:** Link to the Grafana dashboard if you have filled in the configuration in frontline.conf
- **Generate public link:** To create a public link
- **Switch to Summary:** Switch to [summary](#) view for Requests & Groups tabs
- buttons to filter the metrics drawn in the charts area



## Generate Public Links

A public link is a link of the current reports which will be accessible to anyone, without having to log-in to FrontLine. To generate a public link, click on the **Generate public link** button and choose the expiration date of your link.

### Generate public link

Choose the expiration date of your public Link: in 0 days

July 2019						
Su	Mo	Tu	We	Th	Fr	Sa
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

[Generate](#) [Cancel](#)

The maximum allowed lifetime for a public link is 1 year.

Once you have chosen an expiration date, you can proceed by clicking on the generate button.

## Public link created successfully

The Public link Public link has been successfully created, make sure to copy it now as you won't have access to it later.

<http://localhost:3000/#/p>

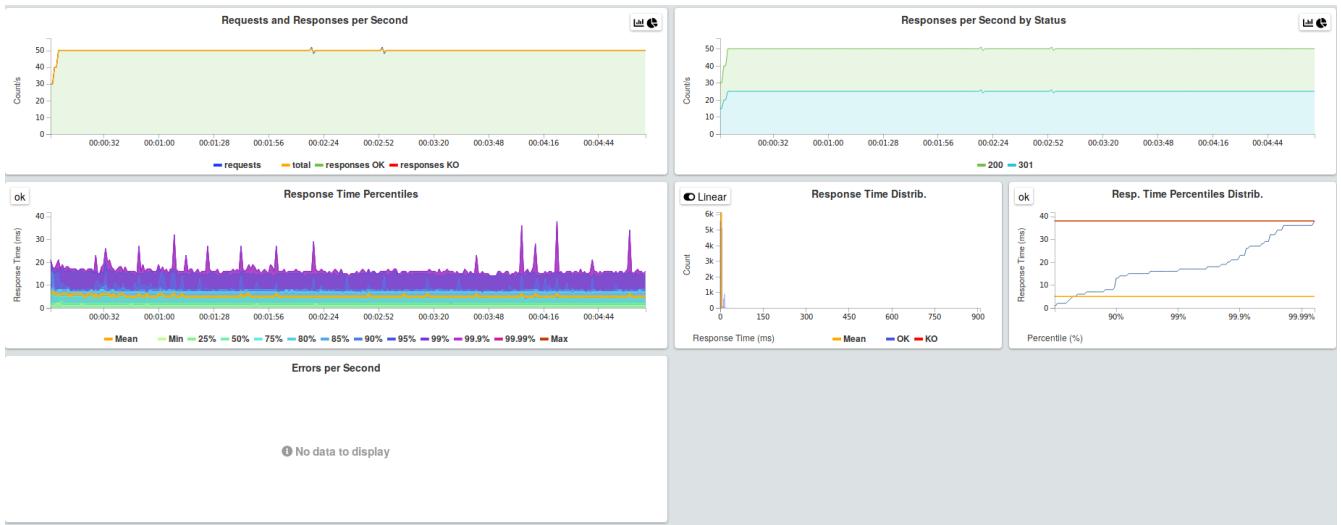
[Copy to clipboard](#)

[OK](#) [Go](#)

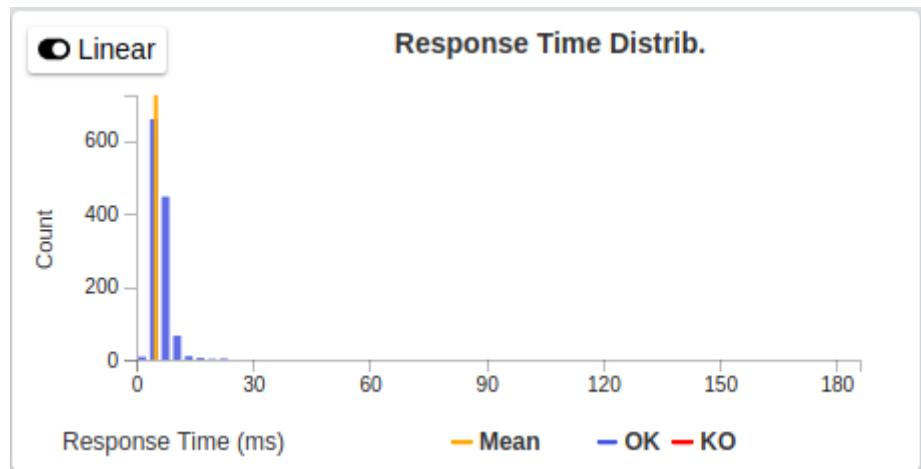
You can copy the public link to share your reports to non-FrontLine users, or click on the "Go" Button to access it yourself. You can click on the "OK" button to close this modal.

## Charts

Each charts in FrontLine are connected to each other, so if you select a time window on a chart it will automatically change it for all other charts. Metrics are drawn in multiple charts.



Some of them have an icon to update the chart settings:



Moreover, histograms and pies are hidden behind each counts charts, accessible by clicking their top right corner icon below.



If your kernel version is too low (around below 3.10) you might not be able to get data from the TCP connection by state graph on the Connections tab. If you want to be able to get these data, you should upgrade your kernel.

## Summary (Requests and Groups only)

This view is available only from requests and groups tabs. It is a summary of metrics drawn in the charts, and has two modes: flat, by default, and hierarchy. The summary is also connected to the timeline and the time window selected, so if you change the time window the summary will refresh his data to match the time window.

On Flat mode you can filter the data by clicking any column name of the table.

The screenshot shows a table titled "Outgoing Counts" and "Incoming Counts" with columns for Total, OK, KO, % KO, and response times (Min, 50%, 75%, 90%, 95%, 99%, 99.9%, 99.99%). The table includes rows for Global, / Search, / home Redirect 1, / home, and / Select.

#	Requests	Outgoing Counts				Incoming Counts				Response Time (ms)									
		Total	/s	Total	OK	KO	% KO	/s	Min	50%	75%	90%	95%	99%	99.9%	99.99%	Max	Mean	Std Dev
-	Global	1 200	18.35	1 200	1 200	0	0	18.35	1	5	6	8	9	14	22	23	23	6	2
3	/ Search	300	4.84	300	300	0	0	4.84	1	7	8	10	11	16	23	23	23	7	2
2	/ home Redirect 1	300	4.84	300	300	0	0	4.84	1	5	5	6	7	11	14	14	14	5	1
1	/ home	300	4.84	300	300	0	0	4.84	1	5	6	8	9	13	22	22	22	6	2
4	/ Select	300	4.84	300	300	0	0	4.84	1	4	5	6	7	9	18	18	18	5	1

## Export PDF

When clicking on the green button in the navigation bar, you will have access to a page where you can configure and then export a PDF report of a specific simulation.

This screenshot shows the Gatling interface with the title "Simulations > IdleTimeoutSimulation > #3". A green "Export" button is visible on the right side of the header.

This report is initialized with:

- a title element with the date of the run you were coming from
- the run status
- the run comments
- the run assertions
- the run requests summary
- 3 charts of the run:
  - Requests and Responses per second
  - Responses per Second by Status
  - Response Time Percentiles

This screenshot shows the "Export to PDF - Sample Simulation" configuration page. It displays three expandable sections: "Title" (containing the date 4/23/2019, 11:07:56 AM - 2m 20s), "Status" (containing the message "Run 296 ended with status: Assertions failed"), and "Comments" (containing the message "There are no comments for the run 296"). At the top right are buttons for "Save..", "Load", and "Export PDF".

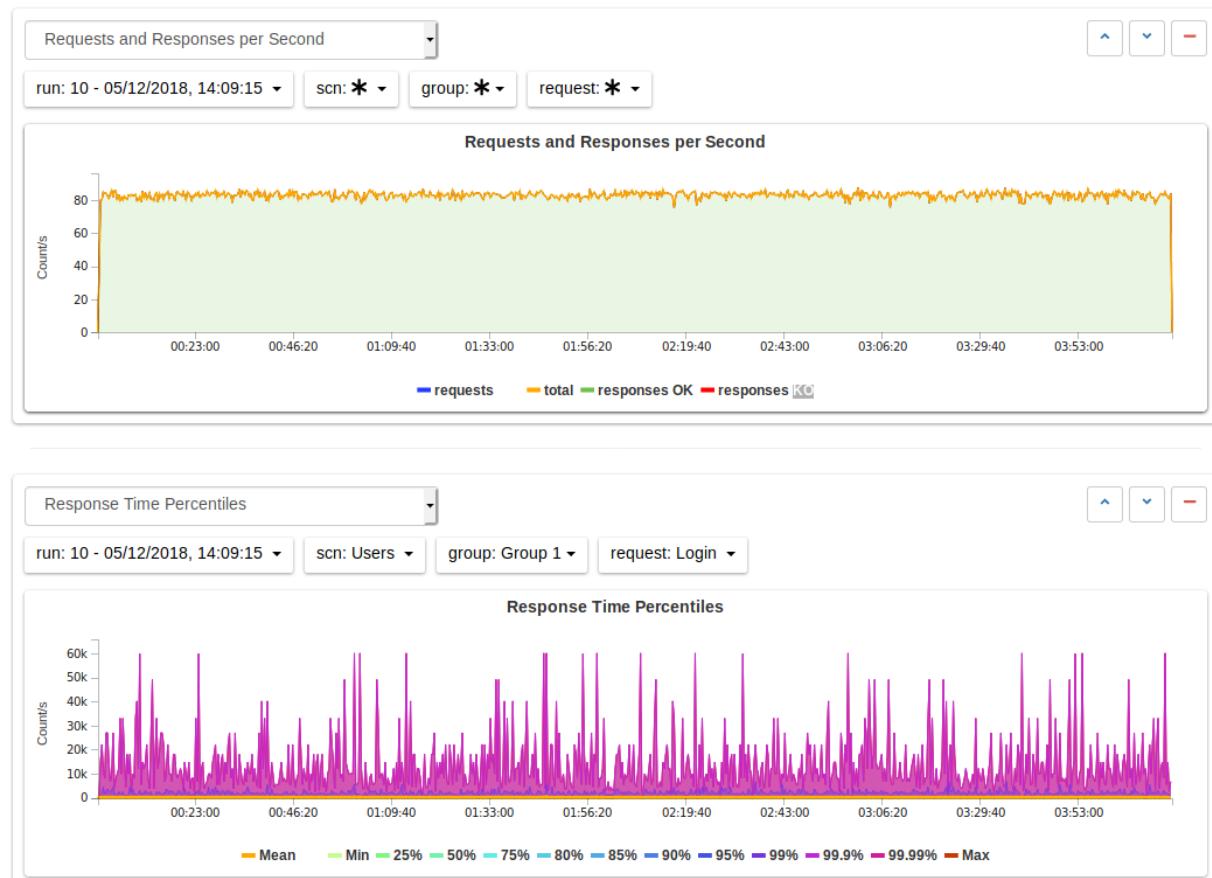
This page is a configurable list of different elements that will be displayed in the report. You can click on the blue add button under every element to add another one.

Every element can be moved up or down by clicking on the blue arrow on the top right of the element, or be removed by clicking on the red dash.

Those elements are composed of:

- **Title:** add a title element.
- **Text Area:** add an editable text element.
- **New Page:** allow you to skip a page in the report.
- **Run:**
  - **Status:** add an editable text element with a predefined text set to the status of the selected run.
  - **Comments:** add an editable text element with a predefined text set to the comments of the selected run.
  - **Assertions:** add a table with the assertions of the selected run.
  - **Summary:** add the summary table of the selected run in a new landscape page.
- **Chart:** add a chart element that you can interact with before exporting it to PDF.
- **Counts:** add a count chart element that you can interact with before exporting it to PDF.

As you can see below, every charts (or other elements) can be interact with individually. You can zoom on it, or select the run, the scenario, the group, etc.. whose you want your data to be fetch. You do not need to have the same settings for each element.



After adding all desired elements in the report you can click on the **Export PDF** button on the top

right to get your PDF file.



There are two more actions you can do:

- **Save:** save the current Export configuration:
  - **as a template:** this option will save the element list without the content
  - **as a save:** this option will save everything, including the content of the Text Area and the configuration of the graphs
- **Load:** load a previously saved template or save.

## Useful Tips

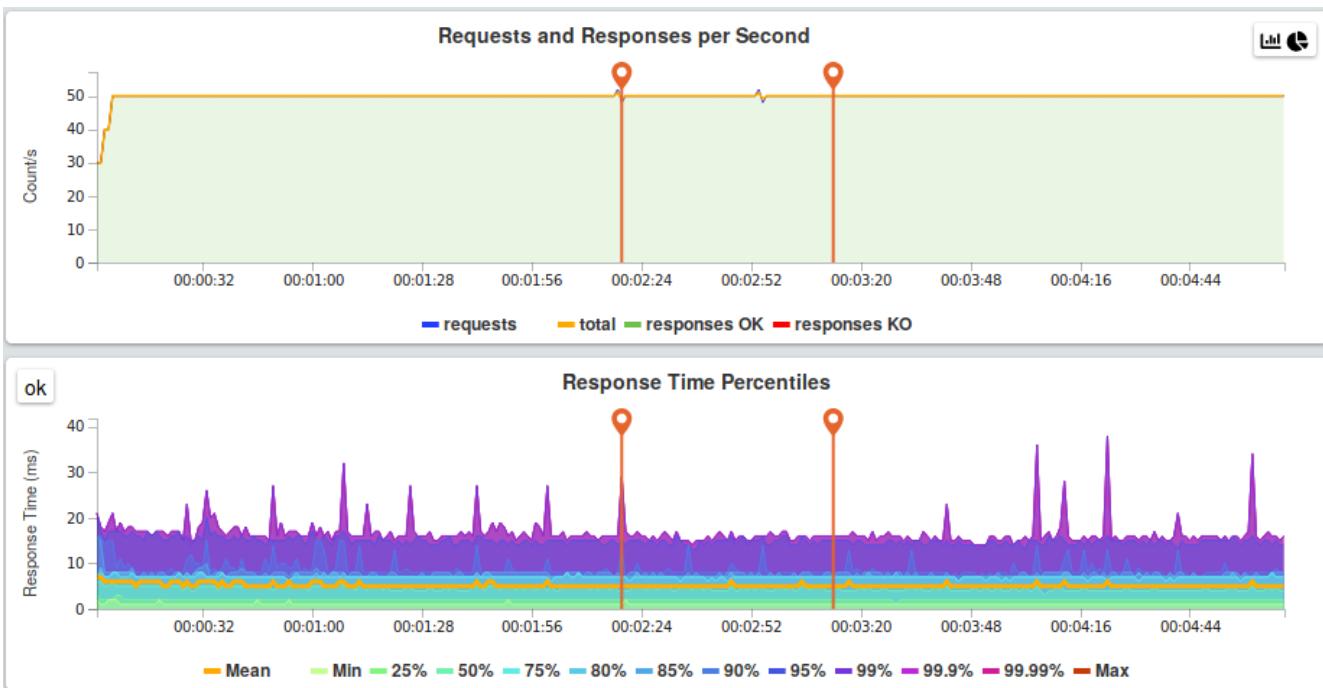
### Zoom

You can reset zoom by double clicking on a chart. It is possible to change the time range window by the following actions:

- Clicking zoom icons of the control buttons
- Select a zone in any charts and timeline
- Select a range time from the top navigation bar

### Markers

To ease your analysis, you can create markers on all the charts by right clicking on them. And click on the top of the marker to delete it.



## Multiple Highlights

In the top right menu, you can activate the **Multiple Highlights** setting which allows the tooltip to be displayed on every chart at the same time.

⌚ Jul 25, 2018 1:48:43 PM → Jul 25, 2018 1:49:24 PM (41s) ▾ Preferences ▾

Duration: 41s ✓ Successful boat

Multiple highlight  Date Time

Resolution: 1 second

Requests Groups Users Connections DNS Bandwidth Injectors

Start Generate public link Switch to Requests Summary scenario: Users group: — request: get

**Requests and Responses per Second**

Counts

00:00:03 00:00:07 00:00:11 00:00:15 00:00:19 00:00:23 00:00:27 00:00:31 00:00:35 00:00:39

— requests — total — responses OK — responses KO

**Responses per Second by Status**

Counts

00:00:03 00:00:07 00:00:11 00:00:15 00:00:19 00:00:23 00:00:27 00:00:31 00:00:35 00:00:39

— 200

**Response Time Percentiles**

ok

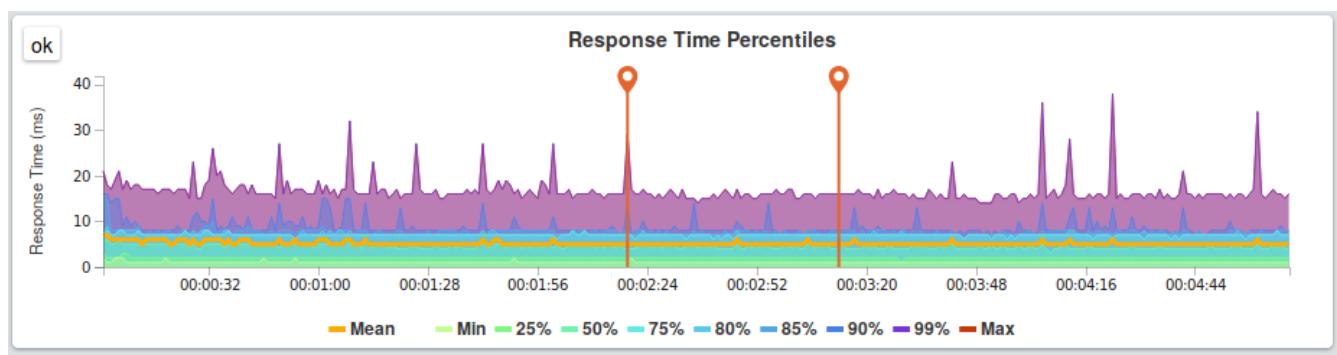
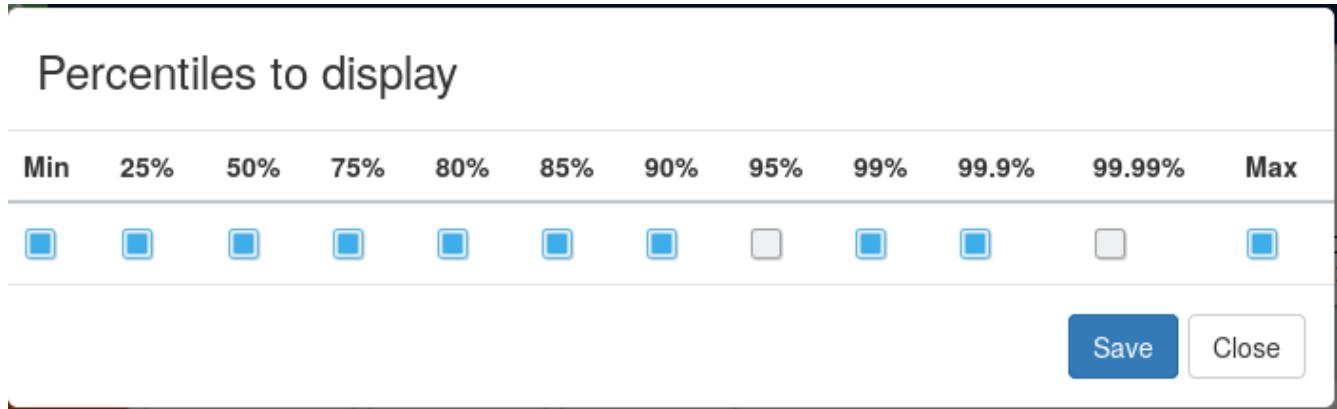
Response Time (ms)

00:00:03 00:00:07 00:00:11 00:00:15 00:00:19 00:00:23 00:00:27 00:00:31 00:00:35 00:00:39

— Mean — Min — 25% — 50% — 75% — 80% — 85% — 90% — 95% — 99% — 99.9% — 99.99% — Max

## Percentiles Mask

In the top right menu, you can click on the **Percentiles** setting to be able to chose what percentiles to display in the chart.

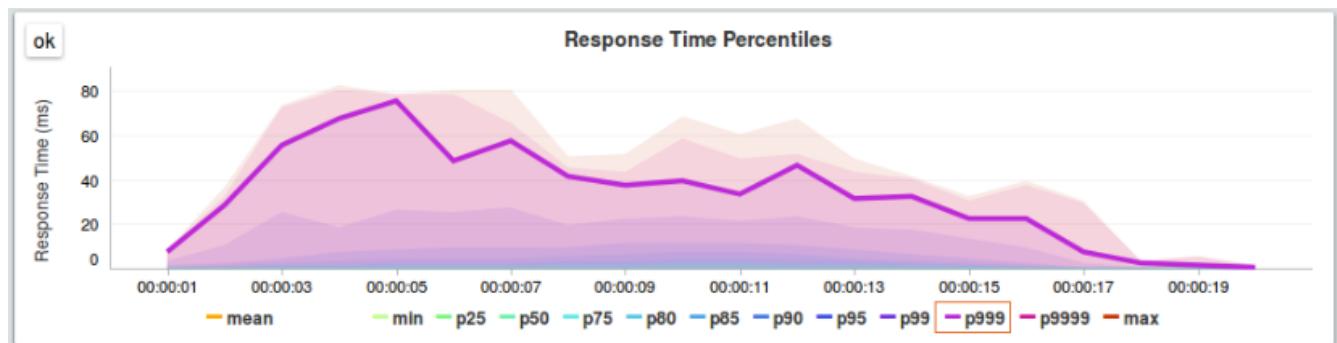


## Date Time / Elapsed Time

In the top right menu, you can activate the **Date Time** setting to be able to switch from elapsed time to date time.

## Highlight Legend

By hovering the label on the percentiles chart legend, you will be able to highlight the curve on the chart, leading to a better view of that curve. The highlight legend options is enable to every "non stacked" graph.



## 2.8. Documentation

You can click on the Documentation icon in the navigation bar on the bottom left of the screen to display every link to the FrontLine documentation and samples.

## Documentation

---

[FrontLine Release Notes](#)  
[FrontLine Installation Guide](#)  
[FrontLine User Guide](#)  
[FrontLine Plugins Guide](#)  
[FrontLine API Documentation](#)  
[FrontLine Samples](#)

---

OK

## 2.9. Plugins Download

If you want to download one of your official FrontLine plugin, please click on the Plugins icon in the navigation bar. For more informations about the plugins, please refer to our plugins guide: <https://gatling.io/docs/frontline/FrontLine-Plugins-Guide.pdf>

### Plugins

---

[Download FrontLine Bamboo Plugin](#)  
[Download FrontLine Jenkins Plugin](#)  
[Download FrontLine TeamCity Plugin](#)  
[Download FrontLine Grafana Plugin](#)  
[FrontLine API Documentation](#)  
[FrontLine Samples](#)

---

OK

## 2.10. About

You can click on the About icon in the navigation bar to display the informations about your FrontLine version and about your license.



**FrontLine Version:** 1.12.0

**Gatling Version:** 3.4.0

**License Expiration Date:** December 21, 2020

**Maximum Number of Simulations:**  $\infty$

**Maximum Number of Load Injectors:**  $\infty$

**Maximum Number of Teams:**  $\infty$

---

OK

# Chapter 3. Public APIs

The FrontLine API server also exposes a public API that you can use to trigger runs or fetch run results and metrics.

We also use this API for our Continuous Integration plugins and our Grafana datasource.

This API is protected with tokens generated by FrontLine administrators through the [Admin > API Tokens](#) tab.

You can access the Swagger documentation using the link [FrontLine API documentation](#) on the Documentation modal accessible on the sidenav (bottom left of the screen).

Some information before using the public API:

- Simulations ID are available on the Simulations Table, at the left of the simulation name.
- Runs ID are available on the Runs Table: [Simulations > Runs history](#). To copy the id, click on the  icon at the right of the run number.
- You can retrieve a team ID by clicking on the ID column inside the teams table.
- You have to provide the run ID as a query parameter to fetch other run metadata (injectors, remotes, hostnames, scenarios, groups, requests)
- The `from` and `to` query parameters from the [/series](#) endpoint are the lower and upper timestamp bounds of the time window you want to query. You can fetch the total run time window from the [/runs](#) endpoint (injectStart, injectEnd).