Alain Cancel
Canal+

Stéphane Landelle
Gatling

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

OPTIMISER LA QUALITÉ ET LA PERFORMANCE DE L'APPLICATION CANAL+ GRÂCE AUX TESTS DE CHARGE END-TO-END AVANCÉS

17 Juin 2025

BEFFROI DE MONTROUGE

CFTL
Comité Français des Tests Logiciels

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Speakers

CANAL+

Alain Cancel
Canal+ Tech - Qualité - Validation
Responsable Validation / Team Leader

Gatling

Stéphane Landelle
CTO & Co-Founder

CANAL+

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

CANAL+, c'est :

- Historiquement, en 1984, une chaîne de TV cryptée Française déployée sur un décodeur

- Aujourd'hui, à 40 ans, un groupe de média et de divertissement mondial

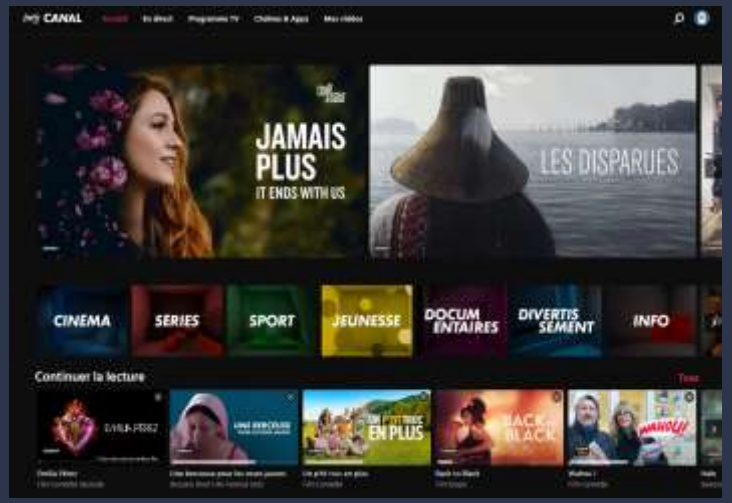
26,4 millions d'abonnés dans le monde
+ 400 millions d'utilisateurs actifs mensuels
52 pays (Europe, Afrique, Asie)

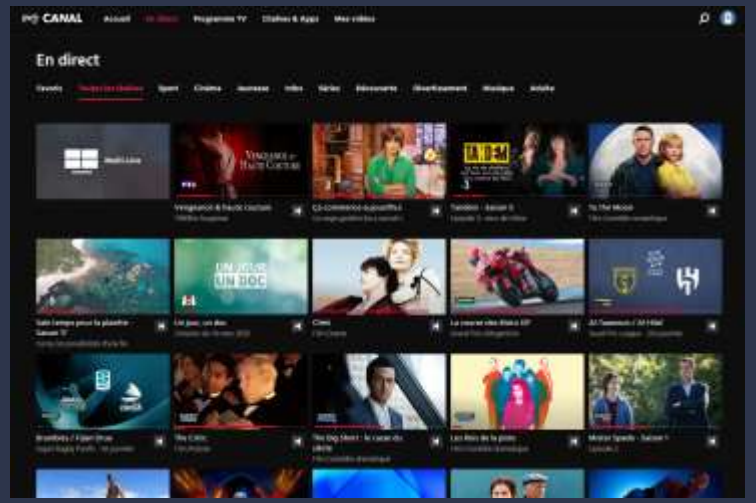
Agrégateur de contenus

Production de contenus
+9400 titres, Catalogue de **Studiocanal**, l'un des plus prestigieux et le plus grand d'Europe

Des évènements majeurs

CANAL+ : L'application CANAL+ < LES FRONTS >





<u>Les Fronts</u> :

Catalogue de contenu : LIVE – VOD
Multi-devices: PC, WEB, MOBILE, TV, DÉCODEUR, CONSOLE ...
Multi-Environnement : Windows, iOS, Android, OneCore, OnePlayer, …
Authentification / Navigation / Consommation de contenu Média

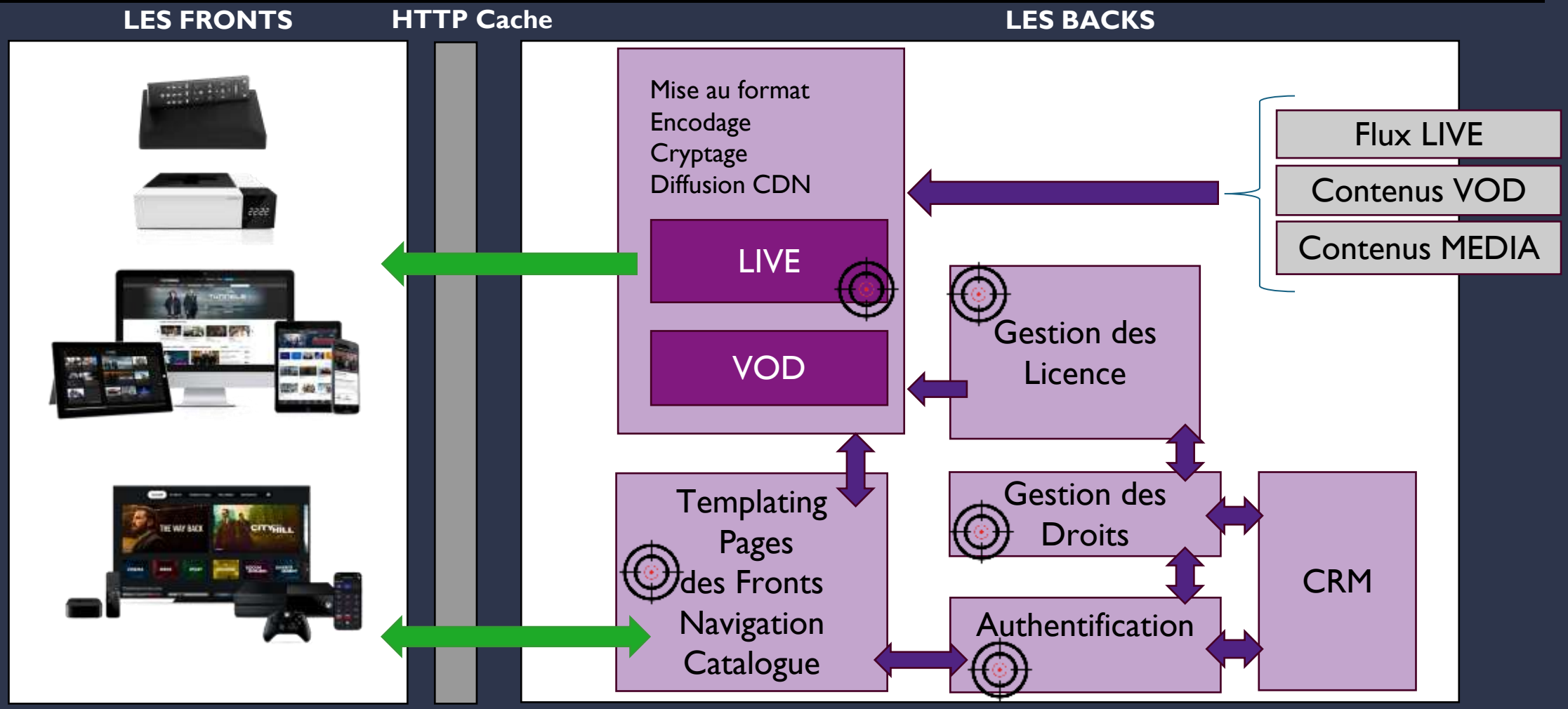
<u>Nos engagements</u> :

Besoin de garantir une continuité de service à nos clients >> une interruption de service de 10s est un incident critique

Gestion des pics d'audiences selon événementiel (un match, une course, un film, une série, une émission, ….)

Gestion des pics de charge selon comportement massif de nos abonnés (connexion, zapping, abonnement, …)

CANAL+ : L'application CANAL+ < LES BACKS >

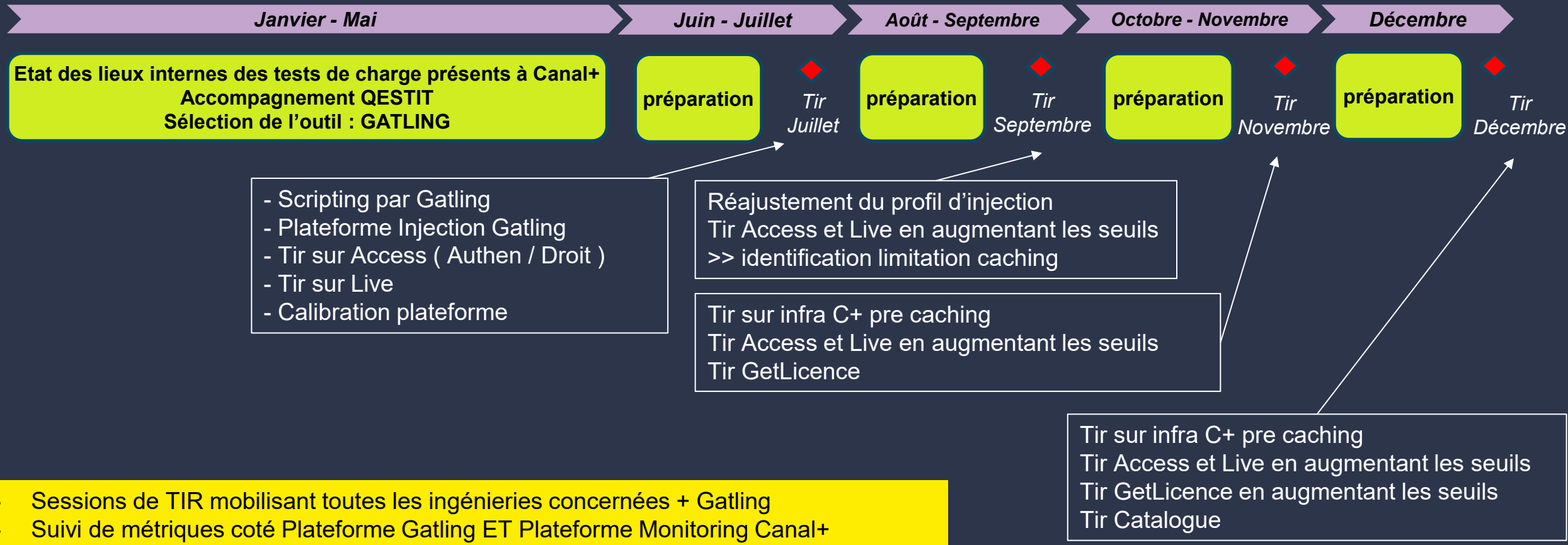


- Autant d'ingénieries que de Backends
  - Authentification / Droit / GetLicence
  - Navigation / Catalogue
  - LIVE / VOD
- Engagement et prévision de charge (régime pic audience / incident évènement) : plusieurs fois le seuil nominal
- Soirée Multiplex : Plusieurs millions utilisateurs en simultanée // Audience C+Foot plusieurs millions

CANAL+ : Test de charge, mais comment ?

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

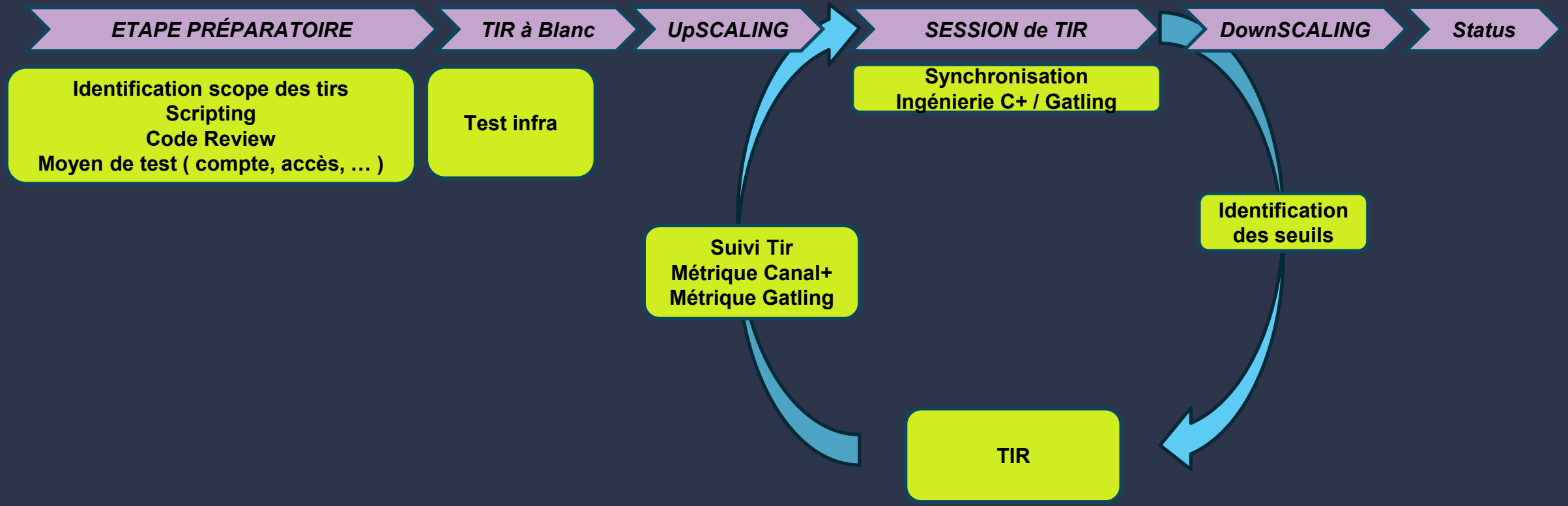
1) Etat des lieux internes
2) Accompagnement par un expert QESTIT
3) Sélection d'un outil
4) Mise en place d'une méthodologie
5) Identification des tirs à réaliser – scripting des scénarios
6) Identification de l'infrastructure de tir



- Sessions de TIR mobilisant toutes les ingénieries concernées + Gatling
- Suivi de métriques coté Plateforme Gatling ET Plateforme Monitoring Canal+
- Sessions de TIR découpées en tirs successifs avec montée en charge progressive

CANAL+ : Déroulé d'une session de tir

JOURNÉE FRANÇAISE DES TESTS LOGICIELS



- Une phase préparatoire
- Une session de tir avec des tirs itératifs en augmentant les seuils selon analyse des résultats précédents

GATLING: introduction

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

FR 📍 **Gatling Corp, Société française, basée à Paris, France.**

💁‍♂️ **40 Collaborateurs**, comprenant toujours l'équipe fondatrice, dont les investisseurs sont français (dont ex-CTO de PriceMinister (Rakuten)).

🌎 **85% du chiffre d'affaires à l'international.** 15% en France.

Test-as-code for load testing.
All GUI-based. Not really DevOps-ready.

Cloud-Native Hybrid SaaS.
On-demand SaaS. Multi-cloud. Self-managed.

Load testing is technical.
Needs more accuracy. Needs more flexibility.

Most flexible platform.
DevOps. Observability. Automation. Integrations.

Tech teams are collaborative.
Needs more openness and integration.

Enterprise-grade Gouvernance.
Security. Data protection. Tech Risk & Compliance.

GATLING : Comment fonctionne un test de charge ?

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Design du Test

Technologie(s) testée(s)

Scénario métier

Profil de charge

Génération du Test

+

Injecteur(s) de charge
1 ou Plusieurs

Traitement des Données

Génère des **Virtual Users** qui réalisent les **Requêtes**

Récupération des **Metrics & Stats** (Counts, Response times, Errors statuses, etc.)

Application/Service
à tester

Restitution & Analyse

Reportings

Scénario métier

Technologie(s) testée(s)

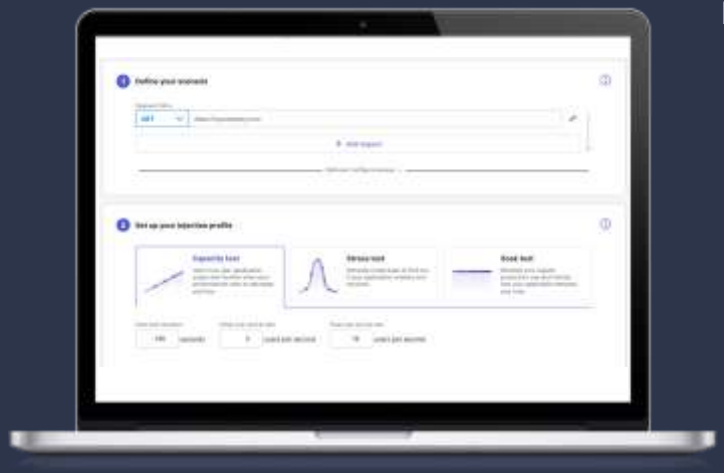
Profil de charge

- Ouvert / Fermé
- Les VUs restent connectés longtemps ou non
- Tests distribués sur plusieurs zones géographiques
- Variabilisation de différentes étapes du parcours

CANAL+ : La création et la modification des tests

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Full No-code
(url-based)



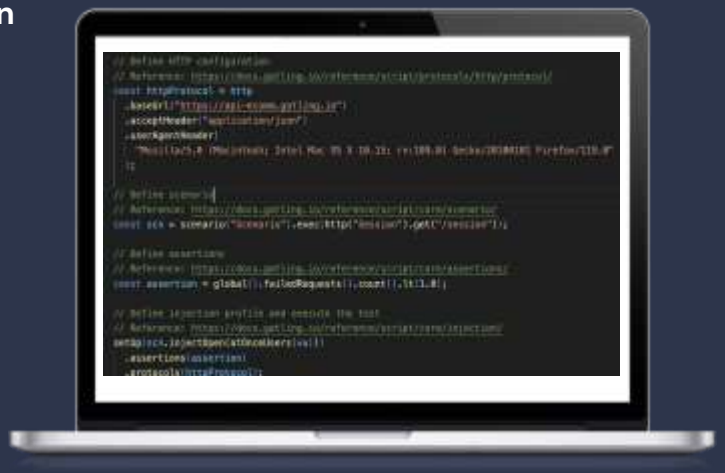
Low-code
(different options)

Browser

Recorder

Postman

Test-as-code
(code-based)



A quoi ressemble un test-as-code ?

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

```java
package example;

import static io.gatling.javaapi.core.CoreDsl.*;
import static io.gatling.javaapi.http.HttpDsl.*;

import io.gatling.javaapi.core.*;
import io.gatling.javaapi.http.*;

public class BasicSimulation extends Simulation {

  // Load VU count from system properties
  // Reference: https://docs.gatling.io/guides/passing-parameters/
  private static final int vu = Integer.getInteger("vu", 1);

  // Define HTTP configuration
  // Reference: https://docs.gatling.io/reference/script/protocols/http/protocol/
  private static final HttpProtocolBuilder httpProtocol = http.baseUrl("https://api-ecomm.gatling.io")
      .acceptHeader("application/json")
      .userAgentHeader(
          "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36");

  // Define scenario
  // Reference: https://docs.gatling.io/reference/script/core/scenario/
  private static final ScenarioBuilder scenario = scenario("Scenario").exec(http("Session").get("/session"));

  // Define assertions
  // Reference: https://docs.gatling.io/reference/script/core/assertions/
  private static final Assertion assertion = global().failedRequests().count().lt(1L);

  // Define injection profile and execute the test
  // Reference: https://docs.gatling.io/reference/script/core/injection/
  {
    setUp(scenario.injectOpen(atOnceUsers(vu)).assertions(assertion).protocols(httpProtocol);
  }
}
```

A quoi ressemble un test-as-code ?

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

```java
package example;

import static io.gatling.javaapi.core.CoreDsl.*;
import static io.gatling.javaapi.http.HttpDsl.*;

import io.gatling.javaapi.core.*;
import io.gatling.javaapi.http.*;

public class BasicSimulation extends Simulation {

  // Load VU count from system properties
  // Reference: https://docs.gatling.io/guides/passing-parameters/
  private static final int vu = Integer.getInteger("vu", 1);

  // Define HTTP configuration
  // Reference: https://docs.gatling.io/reference/script/protocols/http/protocol/
  private static final HttpProtocolBuilder httpProtocol = http.baseUrl("https://api-ecomm.gatling.io")
      .acceptHeader("application/json")
      .userAgentHeader(
          "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36");

  // Define scenario
  // Reference: https://docs.gatling.io/reference/script/core/scenario/
  private static final ScenarioBuilder scenario = scenario("Scenario").exec(http("Session").get("/session"));

  // Define assertions
  // Reference: https://docs.gatling.io/reference/script/core/assertions/
  private static final Assertion assertion = global().failedRequests().count().lt(1L);

  // Define injection profile and execute the test
  // Reference: https://docs.gatling.io/reference/script/core/injection/
  {
    setUp(scenario.injectOpen(atOnceUsers(vu))).assertions(assertion).protocols(httpProtocol);
  }
}
```

1. Import SDK classes
2. Pass dynamic variables
3. Define HTTP Protocol
4. Define Scenario
5. Set acceptance criteria
6. SetUp Injection Profile

Etape 1


```
import static io.gatling.javaapi.core.CoreDsl.*;
import static io.gatling.javaapi.http.HttpDsl.*;

import io.gatling.javaapi.core.*;
import io.gatling.javaapi.http.*;
```

1. Import SDK classes

Etape 2


```
// Load VU count from system properties
// Reference: https://docs.gatling.io/guides/passing-parameters/
private static final int vu = Integer.getInteger(nm:"vu", val:1);
```

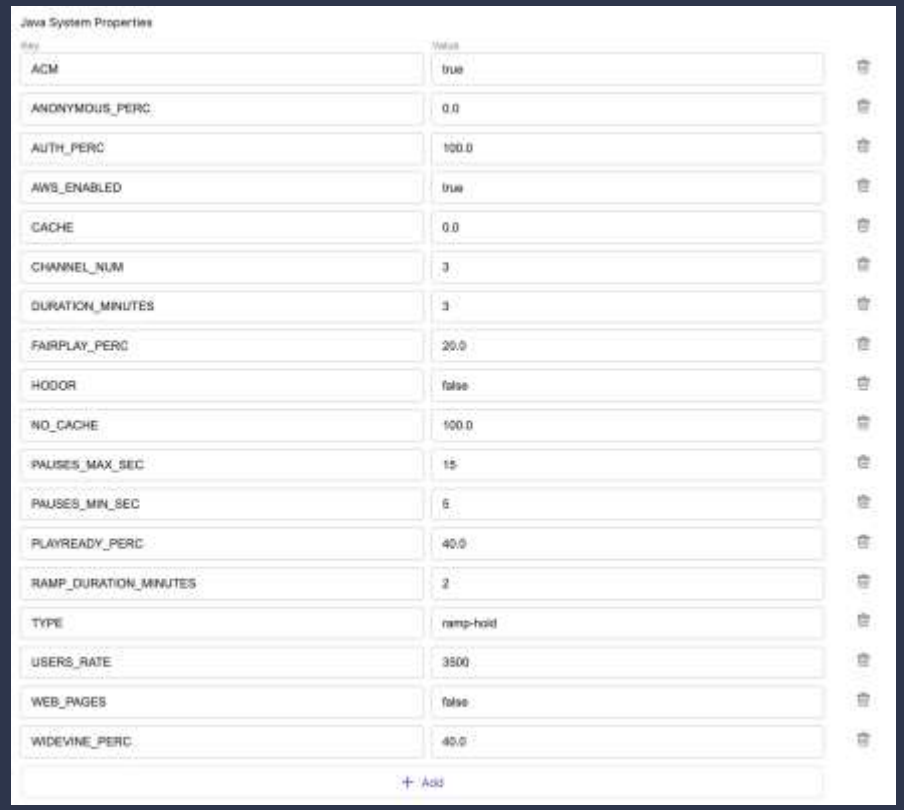
2. Pass dynamic variables

- Modification du comportement du script à la volée à partir de la CLI ou le GUI.
- Pas besoin de modifier le code.

Etape 2

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

- La répartition du trafic sur les différents services
- L'activation des caches
- L'activation du frontend
- Le taux d'arrivée des utilisateurs virtuels
- Le type de test de charge
- La durée des pauses



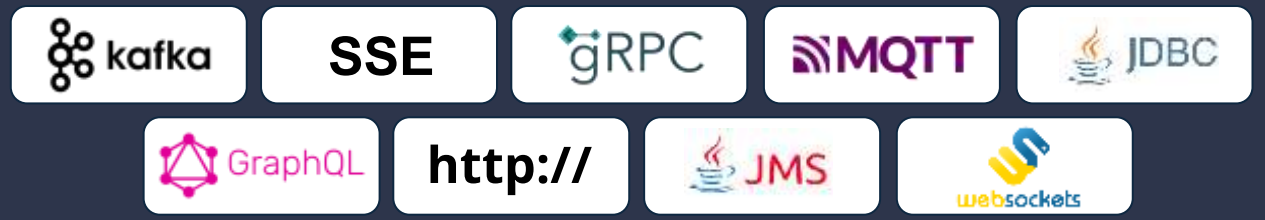
Etape 3


```
// Define HTTP configuration
// Reference: https://docs.gatling.io/reference/script/protocols/http/protocol/
private static final HttpProtocolBuilder httpProtocol = http.baseUrl("https://api-ecomm.gatling.io")
    .acceptHeader("application/json")
    .userAgentHeader(
        "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36");
```

3. Define HTTP Protocol

Configuration de l'agent :

- Base URL
- Headers



JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Etape 3

```
public static HttpProtocolBuilder httpProtocol =
    http.acceptHeader(value:"*/*")
        .acceptEncodingHeader(value:"gzip, deflate")
        .acceptLanguageHeader(value:"fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7")
        .userAgentHeader(
            value:"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36")
        .sign(
            (request, session) -> {
              request.getHeaders().set(name:"gatling-test", value:"1");
              return request;
            });
```

Ajout d'une signature pour identifier, côté serveur (APM), les logs et les métriques provenant du test de charge Gatling

Etape 4

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

```
// Define scenario
// Reference: https://docs.gatling.io/reference/script/core/scenario/
private static final ScenarioBuilder scenario = scenario("Scenario").exec(http("Session").get("/session"));
```

4. Define Scenario

- Le parcours utilisateur à tester
- Intégration des APIs

Etape 4

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

- Évacuation des utilisateurs en échec
- Jeux de données via des feeders
- Groupement de plusieurs requêtes pour créer des briques réutilisables
- Répartition en pourcentage
- Pause de réflexion

```
static ScenarioBuilder mainScn = scenario(name:"MyCanalWeb_scenario")
    .exitBlockOnFail()
    .on(
        feed(usersFeeder),
        setServicesFilter,
        setDeviceIds,
        randomSwitch()
            .on(
                percent(anonymous_perc)
                    .then(
                        group(ANONYMOUS_GROUP)
                            .on(
                                homeAnonymous,
                                pause(minSecond, maxSecond),
                                login,
                                pause(minSecond, maxSecond),
                                homeAuthenticated,
                                pause(minSecond, maxSecond),
                                live,
                                pause(minSecond, maxSecond),
                                loadChannel)),
                percent(auth_perc)
                    .then(
                        group(AUTHED_GROUP)
                            .on(
                                homeAuthenticated,
                                pause(minSecond, maxSecond),
                                live,
                                pause(minSecond, maxSecond),
                                loadChannel))))
    .exitHereIfFailed();
```

Etape 5

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

```
// Define assertions
// Reference: https://docs.gatling.io/reference/script/core/assertions/
private static final Assertion assertion = global().failedRequests().count().lt(1L);
```

5. Set acceptance criteria

Vérifier que le nombre de requêtes en échec est inférieur à 1.

Etape 5


```java
// Define assertions for different test types
// Reference: https://docs.gatling.io/reference/script/core/assertions/
static final List<Assertion> assertions = List.of(
    global().responseTime().percentile(90.0).lt(500),
    global().failedRequests().percent().lt(5.0));

static final List<Assertion> getAssertions() {
  return switch (testType) {
    case "capacity", "soak", "stress", "breakpoint", "ramp-hold" -> assertions;
    case "smoke" -> List.of(global().failedRequests().count().lt(1L));
    default -> assertions;
  };
}
```

Des critères d'acceptation dynamiques qui s'adaptent en fonction du type de test.

Etape 6

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

```
// Define injection profile and execute the test
// Reference: https://docs.gatling.io/reference/script/core/injection/
{
  setUp(scenario.injectOpen(atOnceUsers(vu))).assertions(assertion).protocols(httpProtocol);
}
```

6. SetUp Injection Profile

Des profils d’injection pour simuler la manière dont les utilisateurs virtuels bombardent l’application cible.

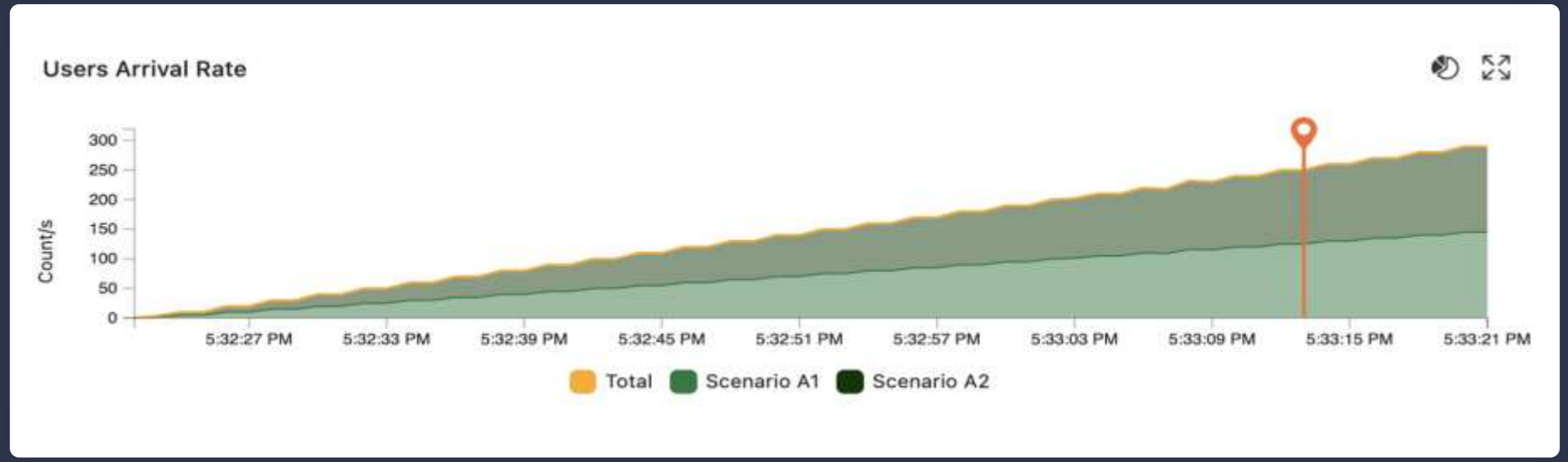
Etape 6

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

- Un profil d'injection dynamique qui s'adapte en fonction du type de test.
- Types :
  - Capacity
  - Soak
  - Stress
  - Breakpoint

```
// Define different load injection profiles
// Reference: https://docs.gatling.io/reference/script/core/injection/
static final PopulationBuilder injectionProfile(ScenarioBuilder scn) {
  return switch (testType) {
    case "capacity" ->
      scn.injectOpen(
          incrementUsersPerSec(vu)
              .times(4)
              .eachLevelLasting(duration)
              .separatedByRampsLasting(4)
              .startingFrom(10));
    case "soak" -> scn.injectOpen(constantUsersPerSec(vu).during(duration));
    case "stress" -> scn.injectOpen(stressPeakUsers(vu).during(duration));
    case "breakpoint" -> scn.injectOpen(rampUsers(vu).during(duration));
    case "ramp-hold" ->
      scn.injectOpen(
          rampUsersPerSec(0).to(vu).during(ramp_duration),
          constantUsersPerSec(vu).during(duration));
    case "smoke" -> scn.injectOpen(atOnceUsers(1));
    default -> scn.injectOpen(atOnceUsers(vu));
  };
}
```

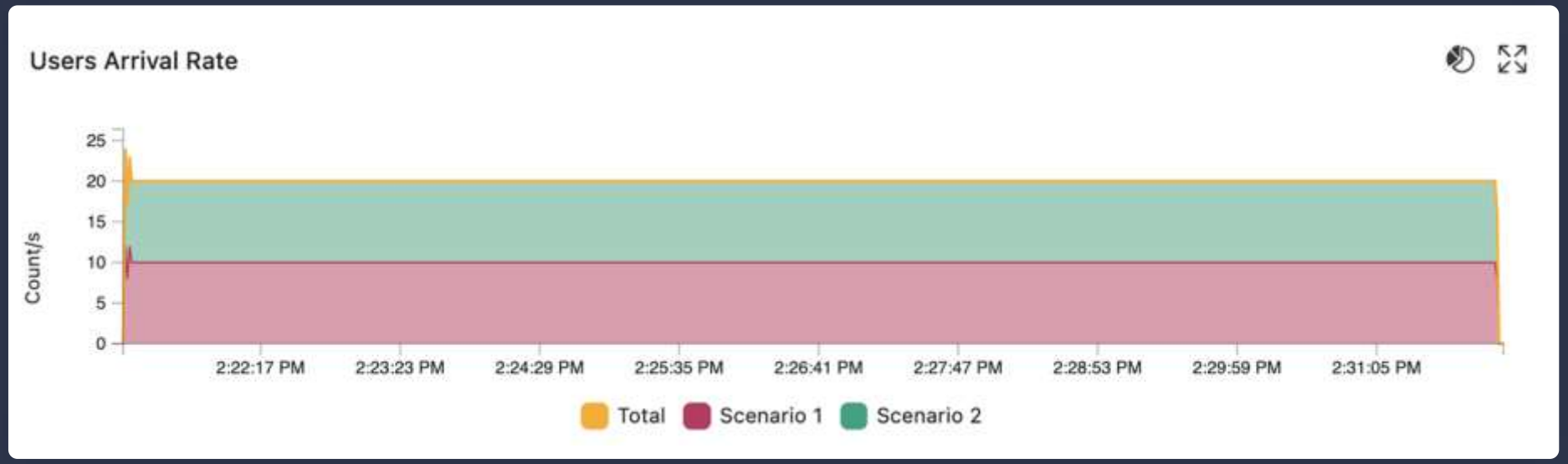
Etape 6: Ramp-up



Tester comment l'application va scaler

Etape 6: Soak Test (Endurance)

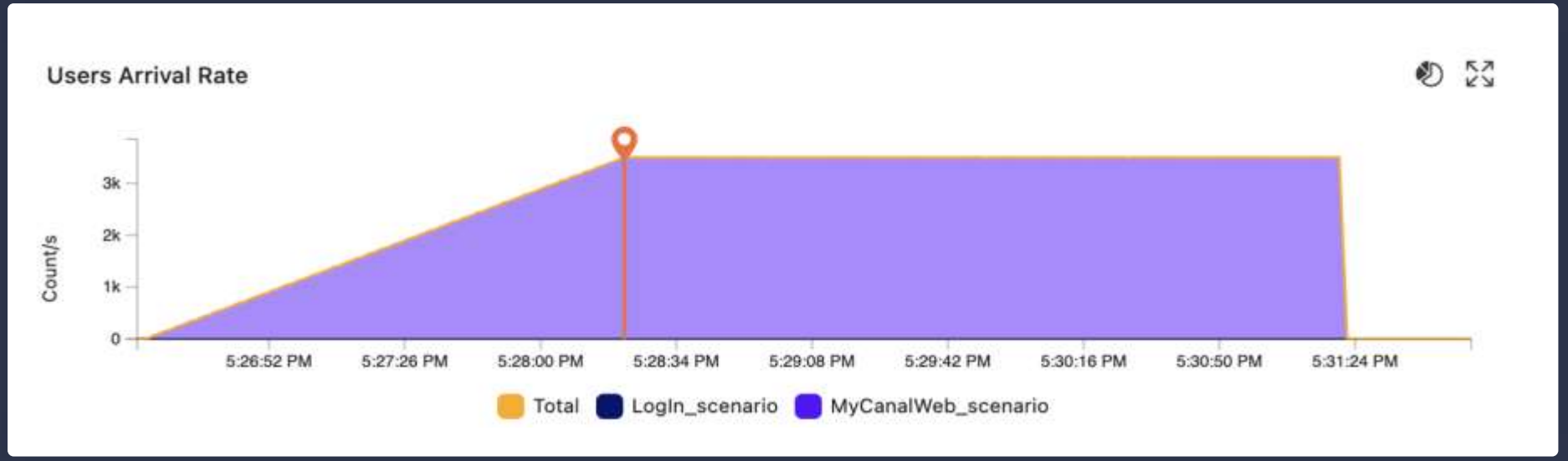
JOURNÉE FRANÇAISE DES TESTS LOGICIELS



Identifier les dégradations de performances au fil du temps.

Etape 6: Soak Test (Endurance)

JOURNÉE FRANÇAISE DES TESTS LOGICIELS



Reproduction du comportement d'arrivée des utilisateurs en production lors d'un match de la Ligue des champions.

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Capacités Test-as-code

- Feeder (Datasource)
  - CSV
  - JSON
  - Sitemap
  - Cache (Redis)
  - Database (MySQL, PostgreSQL, MariaDB, Oracle, etc.)
- Extraction des réponses
  - Validations
  - Corrélations des requêtes
  - Transformation
- Config-as-code
  - Configurer des packages et des simulations sans utiliser l'interface de Gatling Enterprise

CANAL+ : L'architecture et l'environnement des Tests

Automation

GitHub · GitLab · Jenkins · AWS CodeBuild · Azure DevOps · TC

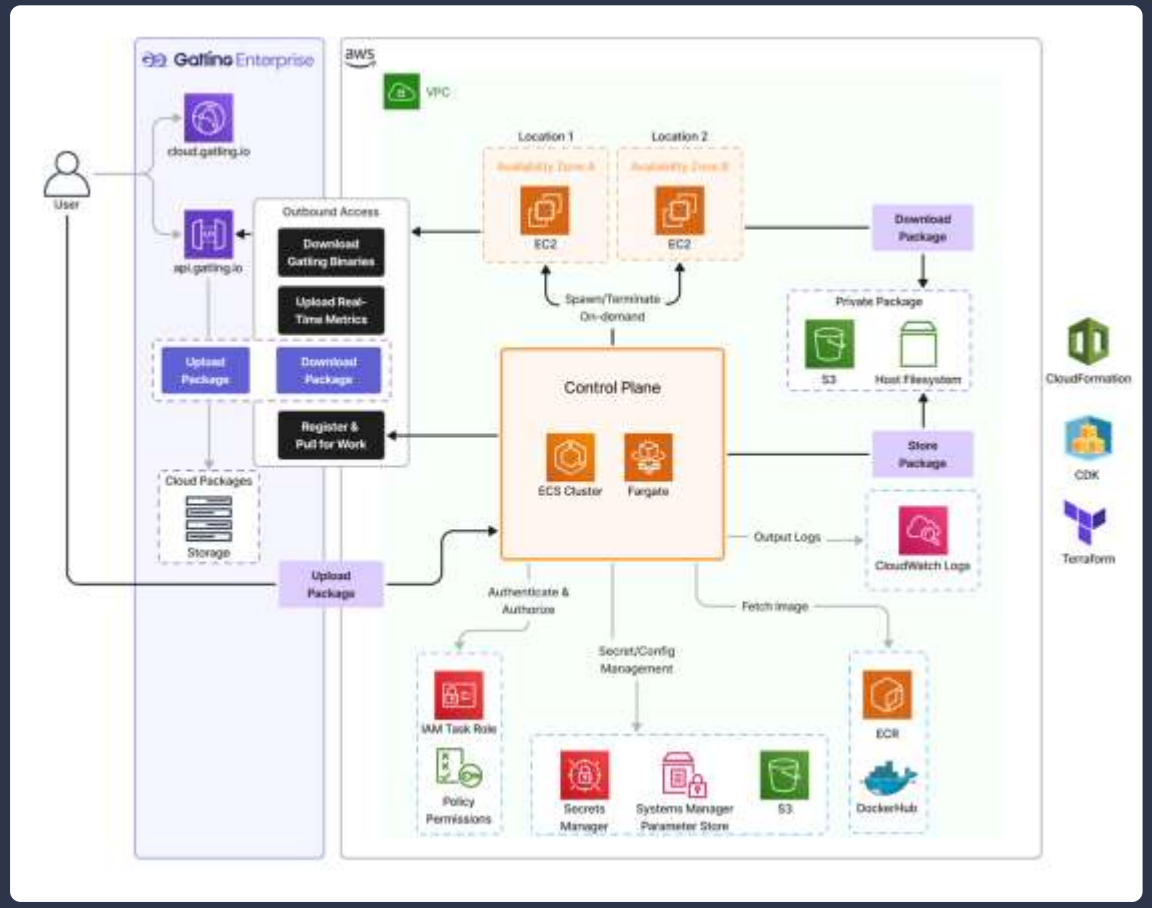
Cloud & Self-hosted

aws · Azure · kubernetes · Google Cloud · RED HAT OPENSHIFT

Infrastructure-as-code

HELM · Terraform · AWS CloudFormation · Cloud Development Kit

- Inclure les tests de charge dans notre cycle DevOps et favoriser l'approche shift-left.
- Lancer des load generators à la demande dans un réseau privé grâce à l'agent Gatling Enterprise.
- Automatiser le déploiement de l'agent Gatling Enterprise avec des configurations Terraform standardisées.



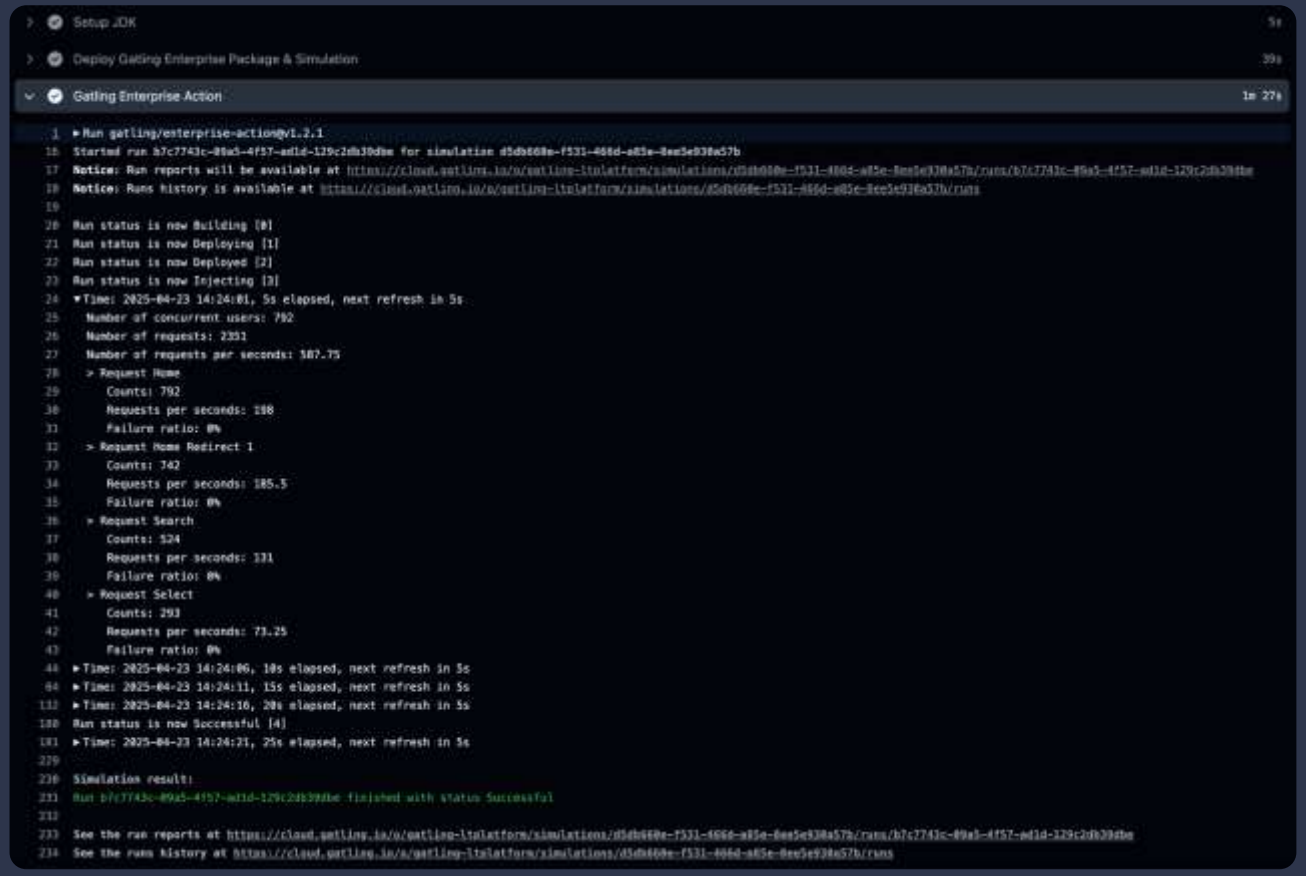
Automatisation des tests


```
provider "aws" {
  region = "eu-west-3"
}

module "location1" {
  source          = "git::https://github.com/gatling/gatling-enterprise-control-plane-deployment//terraform/aws/location"
  id              = "prl_aws_1"
  region          = "eu-west-3"
  subnets         = ["subnet-0b01f98a2d83dd366"]
  security-groups = ["sg-05abe44cd13e8c517"]
  instance-type   = var.instance_type
  elastic-ips     = var.location1_ips
  auto-associate-public-ipv4 = false
}

module "location2" {
  source          = "git::https://github.com/gatling/gatling-enterprise-control-plane-deployment//terraform/aws/location"
  id              = "prl_aws_2"
  region          = "eu-west-3"
  subnets         = ["subnet-08e4b7c460196ca0f"]
  security-groups = ["sg-05abe44cd13e8c517"]
  instance-type   = var.instance_type
  elastic-ips     = var.location2_ips
  auto-associate-public-ipv4 = false
}
```

```
module "control-plane" {
  source           = "git::https://github.com/gatling/gatling-enterprise-control-plane-deployment//terraform/aws/control-plane"
  name             = "control-plane"
  token-secret-arn = var.token_secret_arn
  subnets          = ["subnet-0b01f98a2d83dd366", "subnet-08e4b7c460196ca0f", "subnet-024cc73b7a25ed12a"]
  security-groups  = ["sg-05abe44cd13e8c517"]
  locations        = [module.location1, module.location2, module.location3]
}
```

Automatisation des tests

- Des logs détaillés
- Des métriques minifiés.
- Retour du résultat de la simulation.



CANAL+ : Résultats et Perspectives

JOURNÉE FRANÇAISE DES TESTS LOGICIELS

Résultats :

- Disponibilité de solutions pour tester la résilience de la plateforme
- Constat de l'augmentation de nos seuils de tir de charge à chaque tir
- Identification de potentiels améliorations à apporter dans la plateforme de l'application CANAL+
    - Correction, paramétrage
    - Augmentation des tailles et types de machines et d'instances
    - Gestion des modes dégradés

- Pas de soucis de charge lors des derniers évènements avec pic d'audience important (plusieurs millions d'utilisateurs soirée multiplex)

Perspectives :

- Étendre cette méthodologie de tir sur l'ensemble des fronts
- Étendre le scope jusqu'au composant player (test en charge des CDNs)

Questions ?

JOURNÉE
FRANÇAISE
DES TESTS
LOGICIELS