



# Web Video Converter

## *SambaTech Assignment Solution*

Author: Rodrigo Guimarães – [rodrigo@agileoperations.com.br](mailto:rodrigo@agileoperations.com.br)

### 1. Introduction

Hello there! I know that you have a lot of these assignments to assess and that's why I've put a lot of effort in this documentation trying to make your work easier. I hope I can accomplish that goal and that I can present something different so that it makes this assessment more enjoyable.

In addition to this document I during all development process I've created automated tests (using Test Driven Development approach), which I expect to be useful as an alternative kind of documentation. The automated tests were created using Spock framework, which have a very expressive syntax. I really I hope it can help.

As requested, application was deployed in a cloud computing provider. In our case, we choose Amazon Web Services (AWS).

### 2. Deployment at Amazon Web Services

You can check application running at:

[www.agileoperations.com.br](http://www.agileoperations.com.br)

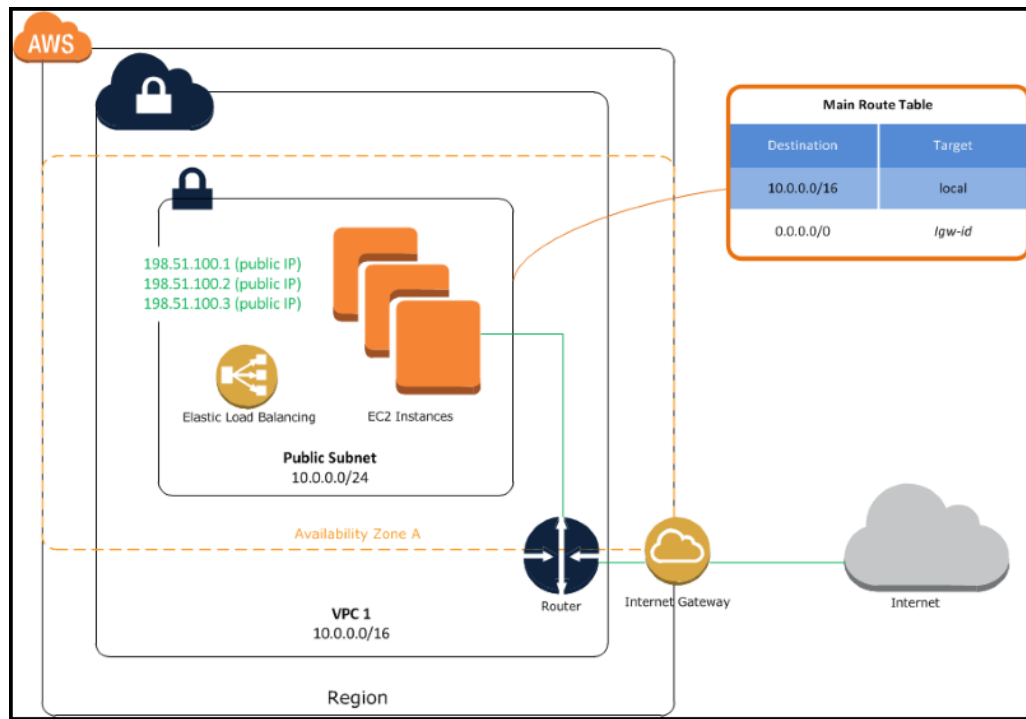
You will need the following credentials to access the web application:

User: admin

Password: admin

Since login for user authentication was not a requirement from the assignment, it is in place to demonstrate capabilities of the system built, not to really enforce security. If security was a concern, other approaches would be taken to restrict access to the application, like creating a strongest password and not sharing everything on public git repository. :)

The DNS to resolve this **agileoperations** domain name is managed by Route53 AWS service. The **Web Video Converter** application was deployed at this address using ElasticBeanstalk (AWS Platform as a Service). This is a managed service that easily instantiate scalable infrastructure for web applications or standalone programs (called workers). In our case, we choose an option from Elasticbeanstalk to deploy the web application following the architectural design depicted in following figure:



You can access the whole infrastructure and check for yourself what was done through AWS console using the following ReadOnly IAM user:

*Thabata from human resources have another version of this document with username and password. Couldn't share this in **public** git repository. In this case, security was a concern for me. Sorry! :(*

<https://agileoperations.signin.aws.amazon.com/console>

You can have more information about Elasticbeanstalk at:

<http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html>

### 3. Code walkthrough

The source code and other artifacts are in github repository:

<https://github.com/loiatan/webVideoConverter>.

It can be cloned executing:

```
$ git clone https://github.com/loiatan/webVideoConverter.git
```

I'll refer to the project root directory (where you can find this README document) as {PROJECT-LOCATION} during the rest of this document.

### **Automated tests**

```
{PROJECT-LOCATION}/test/unit/com/agileoperations/webvideoconverter  
{PROJECT-LOCATION}/test/integration/com/agileoperations/webvideoconverter
```

### **Java Classes**

The java classes are located at:

```
{PROJECT-LOCATION}/src/java/com/agileoperations/webvideoconverter
```

### **Grails Groovy Classes**

The grails groovy classes are used for grails controllers and services. They can be found at:

```
{PROJECT-LOCATION}/webVideoConverter/grails-app/controllers/com/agileoperations/webvideoconverter  
{PROJECT-LOCATION}/webVideoConverter/grails-app/services/com/agileoperations/webvideoconverter
```

There are also some Grails specific view formats (GSP) that can be found at:

```
{PROJECT-LOCATION}/webVideoConverter/grails-app/views/videoConversion
```

In total, five classes were created to solve the problem of this assignment (test classes not counted here):

- **S3Client**
- **ZencoderClient**
- **FileFactory**
- **VideoConversionService**
- **VideoConversionController**

Besides those file classes, two views were created:

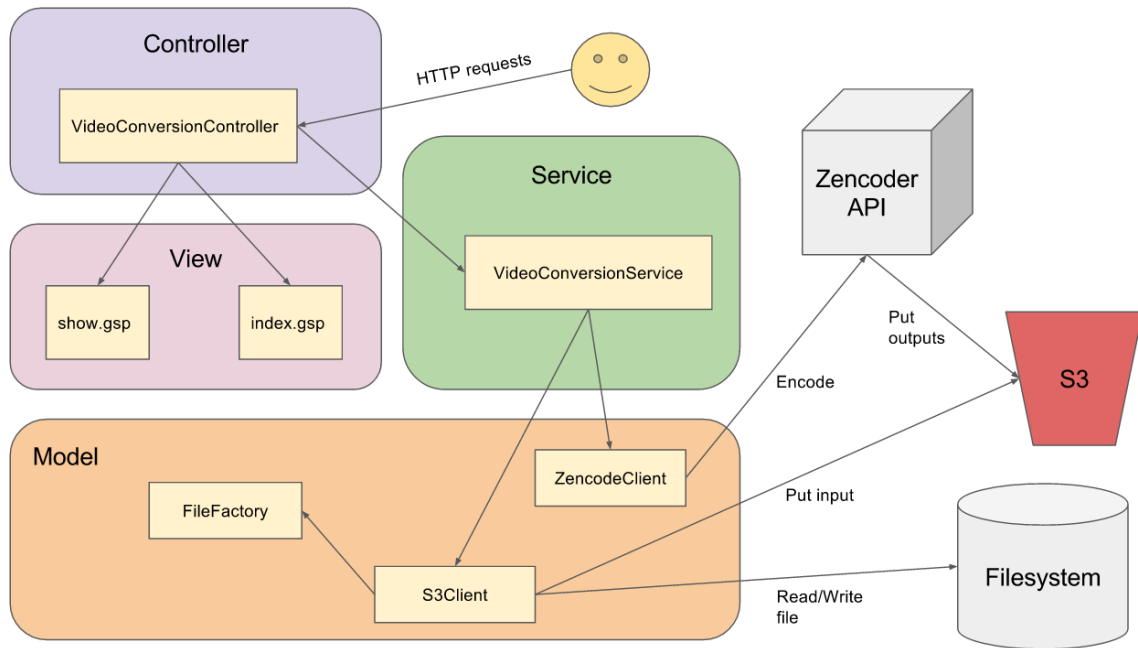
- **VideoConversion/show.gsp**
- **VideoConversion/index.gsp**

Next we explain how these files are connected in a design overview.

## **4. Web Video Converter Design overview**

The design of this solution followed a MVC approach, following framework guidelines, separating responsibilities in the familiar Model, View and Controller layers with the bonus

Service layer here. This idea is depicted in the following figure, connecting the components of the system with external services that they interact (ex: AWS S3 and Zencoder).



The Web application uses **Grails framework** that enforces MVC approach.

This design also helps to write automated tests for the code, which helps on maintainability. Having the code extensively tested makes easier to make a change in the code and know if you break something.

During next steps I'll guide you on how to build and run the program if you are interested to do so.

## 5. Environment to run locally (Optional)

The debian package viewer was crafted using Java 7 Update 75 and Grails 2.4.3 as the framework. To install and build the application you must have installed in your system:

- Java runtime environment 7 Update 75 or newer
- Grails 2.4.x (there are cases that even minor versions are incompatible, so it is suggested that it

### To install the JRE:

1. Go to <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Download the JRE 8 for your system;
3. Execute the installation and follow the instructions given at the screen;

### To run the application:

For example, to run the application you just need to execute:

```
$ grailsw run-app
```

You should see an output like this:

```
-----
Downloading http://dist.springframework.org.s3.amazonaws.com/release/GRAILS/grails-2.4.3.zip to
C:\Users\Rodrigo\.grails\wrapper\grails-2.4.3
.....
.....
| Running Grails application
Configuring Spring Security Core ...
... finished configuring Spring Security Core
| Server running. Browse to http://localhost:8080/webVideoConverter
```

Accessing the link above. You can find additional information about Grails at:

<http://grails.org>

## 6. Running automated tests

To run automated tests you can execute:

```
$ grailsw test-app
```

It is important you have /tmp/ directory created and with permissions to read and write.

If you want to run unit and integration tests separately you can execute:

```
$ grailsw test-app unit:
$ grailsw test-app integration:
```

## 7. War file deployed at AWS Elasticbeanstalk

The war file deployed at Elasticbeanstalk was generated with the following command:

```
{PROJECT-LOCATION}$ grails war
```

The war file is generated inside {PROJECT-LOCATION}/target

During development, several “Releases” were deployed at Elasticbeanstalk to test in our “production” environment. You can check the deployment history using ReadOnly IAM user provided previously.

Environments

Application Versions

Saved Configurations

Delete Deploy Upload Refresh

<input type="checkbox"/>	Version Label	Description	Date Created	Source	Deployed To
<input type="checkbox"/>	webVideoConverter-0.8		2016-04-19 02:06:39 UTC-0400	<a href="#">2016110uOb-webVideoConverter-0.8.war</a>	webvideoconverter-env-vpc
<input type="checkbox"/>	code-pipeline-1461045257511-MyApp-515e400a-9279-4ea1-8d54-c14d8994d89a	MyApp-515e400a-9279-4ea1-8d54-c14d8994d89a	2016-04-19 01:54:18 UTC-0400	<a href="#">webVideoConverterPip/MyApp/87whZhk.zip</a>	
<input type="checkbox"/>	webVideoConverter-0.7		2016-04-18 04:12:59 UTC-0400	<a href="#">2016109van-webVideoConverter-0.7.war</a>	
<input type="checkbox"/>	code-pipeline-1460964141150-MyApp-77fdb4be-1f36-40a1-882b-ad80782a7028	MyApp-77fdb4be-1f36-40a1-882b-ad80782a7028	2016-04-18 03:22:21 UTC-0400	<a href="#">webVideoConverterPip/MyApp/hH2sdta.zip</a>	
<input type="checkbox"/>	code-pipeline-1460945113058-MyApp-6e5fe67a-d6cb-4fcc-8190-2b3cd949d1e4	MyApp-6e5fe67a-d6cb-4fcc-8190-2b3cd949d1e4	2016-04-17 22:05:13 UTC-0400	<a href="#">webVideoConverterPip/MyApp/cxEHZFh.zip</a>	
<input type="checkbox"/>	webVideoConverter-0.5		2016-04-17 20:18:45 UTC-0400	<a href="#">2016109aNY-webVideoConverter-0.5.war</a>	
<input type="checkbox"/>	webVideoConverter-0.4		2016-04-17 18:57:42 UTC-0400	<a href="#">2016108HUz-webVideoConverter-0.4.war</a>	
<input type="checkbox"/>	code-pipeline-1460917946497-MyApp-f2acdbf7-4559-4385-aeb2-	MyApp-f2acdbf7-4559-4385-aeb2-	2016-04-17 14:32:27 UTC-0400	<a href="#">webVideoConverterPip/MyApp/d6</a>	

Of course, those deployments was done just if automated tests passed with success. :)

## 8. The end!

Hope you had some fun like I did doing this assignment! I really appreciate your time on evaluating my solution and I hope that I can be part of your SambaTech team soon! :)

# Big Thank You!

# **Apendix I - Samba Tech - Processo de Seleção Arquiteto de Soluções Cloud**

## Atividade Prática

O desafio consiste em construir e colocar no ar uma aplicação web que possibilita a conversão de arquivos de vídeo de um formato específico, não compatível com padrões da web, para um formato que seja compatível com os padrões da web. A aplicação deve ter uma interface web que permita a inclusão de um novo arquivo de entrada e, após finalizado o processo, permita o usuário assistir o vídeo no navegador. Considere o uso de testes unitários. Atente-se também para o desenho/arquitetura da solução, de modo que a mesma possa ser facilmente escalável, sob padrões de cloud computing. Adicionalmente, a demonstração de habilidades com práticas/ferramentas DevOps também são bem vindas (porém

Itens entregáveis:

- 1) O endereço web (URL) da aplicação funcionando;
- 2) O endereço web do repositório github com o código da aplicação e demais artefatos do projeto disponíveis para análise;

Os entregáveis devem ser submetidos para [rh@sambatech.com.br](mailto:rh@sambatech.com.br).

Regras e orientações importantes

- 1) Para o arquivo de entrada, utilize o exemplo `sample.dv` disponível em

<http://dinamica-sambatech.s3.amazonaws.com/sample.dv>

- 2) Gestão de Código Fonte: Os fontes devem ser disponibilizados em um repositório público do GitHub (<http://github.com>). E será avaliado o histórico dos commits para verificar se você fez commits frequentes e com boa separação temática entre eles. Caso ainda não tenha, você deve criar uma conta e um repositório lá

- 3) Você deve rodar sua aplicação em algum provedor de Cloud Computing dentre os oferecidos na lista abaixo. Todos eles oferecem versões gratuitas que são suficientes para esta atividade;

- 4) Para realizar a conversão do arquivo de entrada no arquivo de saída, utilize também algum dos serviços de encoding oferecidos na lista abaixo. Igualmente estes serviços oferecem opções gratuitas que são suficiente para esta atividade;

- 5) Os arquivos de entrada e de saída devem ficar armazenados no serviço de storage Amazon S3 (também oferecido na lista abaixo).

6) Linguagens: a aplicação deve ser desenvolvida em Java (preferencialmente) ou Python. Qualquer framework disponível para estas linguagens é permitido.

7) Não utilize bibliotecas específicas para acesso ao serviço de encoding. O uso de biblioteca específica para acesso ao serviço de armazenamento da Amazon (S3) é permitido. Lista de Serviços Cloud Computing com Endereços

Para para rodar sua aplicação:

Amazon Web Services - <http://aws.amazon.com>

Microsoft Azure - <http://azure.microsoft.com>

Alternativas:

Heroku: <https://heroku.com>

Google App Engine (Somente Java e Python):

<https://developers.google.com/appengine/>

Serviços de Storage:

Amazon Simple Storage Service (Amazon S3)

<http://aws.amazon.com/s3/>

Microsoft Azure Blob Storage

<http://azure.microsoft.com>

Serviços de Encoding:

Zencoder:

<http://zencoder.com>

<http://zencoder.com/pricing/> (Use o profile "Test" para não pagar)