



Deployment Checklist & Testing Report

# THE QLIKTAG PLATFORM V3.1

Version 1.0

Published Date 10 Aug 2020

Last Revision Date 10 Aug 2020

## DEPLOYMENT CHECKLIST

This document describes the activities performed after production environment setup and deployment is completed. The aim of this is to verify and confirm the environment is functioning as expected.

<b>Date</b>	<b>Verified By</b>	<b>Reviewed By</b>	<b>Comments</b>
10 Aug 2020	Mohit Kumar	Ram Prasad Palugula	

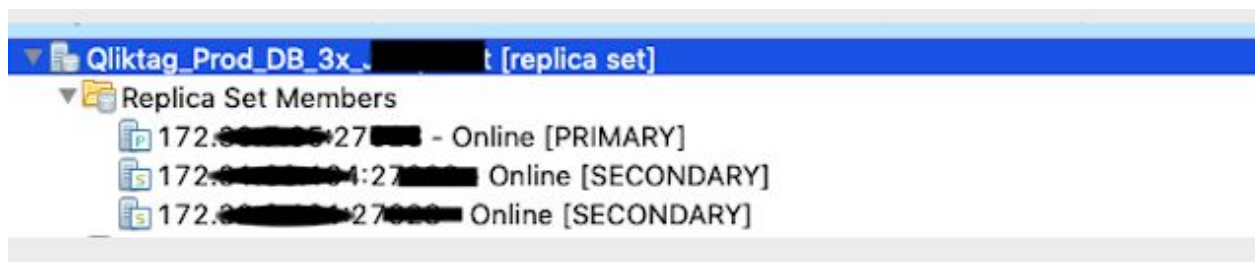
<b>Environment</b>	<b>Description</b>
Production Live	Production environment
Production DR Environment	Disaster Recovery environment
Sandbox	A sandbox environment on same production code version which is used for testing's or pre-production development by enterprise customers

## Production Environment

S No	Review Item	Result (Pass/Fail)
1	Verify the database replica set connection	Pass
2	Verify the Elasticsearch status	Pass
3	Verify the bitbucket pipeline status	Pass
4	Verify lambda service status in AWS console	Pass
5	Verify the lambda service running on NodeJS 10.x	Pass
6	Verify the web application is deployed with Angular correctly	Pass
7	Verify the API health status /api/health	Pass
8	Browse the web application and verify the page loading	Pass
9	Verify the login to the web application	Pass
10	Navigate inside the web application and verify few pages	Pass
11	Test few API GET, POST operations	Pass
12	Verify emails send through application	Pass
13	Verify image uploads through API and web application	Pass
14	Verify few visual Interaction landing pages	Pass
15	Verify the cloudwatch logs	Pass

Below are the proof of screen shots for each review item:

### SNo #1



### SNo #2

## qliktag-production

Deployment name: qliktag-production [Edit](#)

Deployment status: ● Healthy

Deployment version: v7.2.1 [Upgrade](#)

## SNo #3

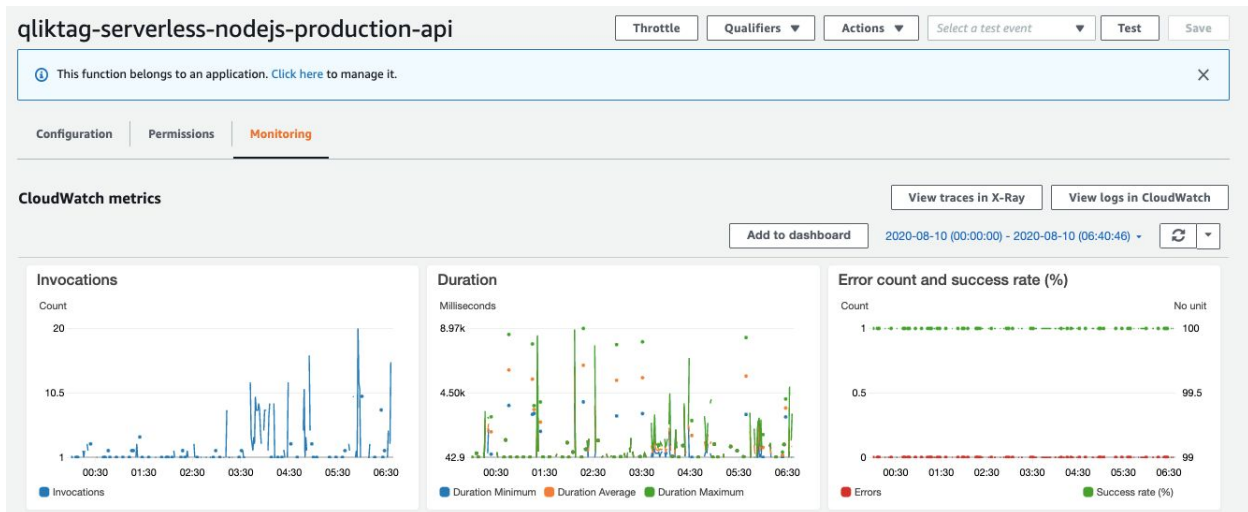
### QliktagWeb application deployment

```
Build
Build setup 38s >
echo "Pipeline Started for Production3x(qliktagweb) enviroment branch" <1s >
node -v <1s >
npm install 12s >
npm run-script build 7m 07s >
aws s3 rm --recursive s3://qliktag-ui-static/production/qliktagweb/ 9s >
aws s3 cp dist/ s3://qliktag-ui-static/production/qliktagweb/ --recursive 14s >
aws s3 rm --recursive s3://qliktag-ui-static-dr/production/qliktagweb/ 17s >
aws s3 cp dist/ s3://qliktag-ui-static-dr/production/qliktagweb/ --recursive 28s >
echo "Pipeline Completed for Production3x(qliktagweb) enviroment branch" <1s >
Build teardown 2s >
```

### QliktagAPI application deployment

```
Build docker
Build setup 27s >
echo "Pipeline Started for Production3x environment branch" <1s >
npm install 10s >
npm install -g serverless 23s >
serverless plugin install --name serverless-webpack 26s >
sls deploy --stage production 4m 34s >
sls deploy --stage production --config serverless-dr.yml 5m 48s >
echo "Pipeline Completed for production3x environment branch" <1s >
echo "Serverless cloudformation deployment is completed successfully" <1s >
apt-get update && apt-get install -y python3-pip 15s >
pip3 install -U awscli 8s >
eval $(aws ecr get-login --no-include-email --region us-east-1) 1s >
docker build -f docker/Dockerfile -t qliktag-3x-container-production . 65s >
echo "Docker image build is completed successfully" <1s >
docker tag qliktag-3x-container-production:latest 781463471155.dkr.ecr.us-east-1.amazonaws.com/qliktag-3x-container-production:latest <1s >
docker push 781463471155.dkr.ecr.us-east-1.amazonaws.com/qliktag-3x-container-production:latest 51s >
echo "Successfully pushed the docker image to repository" <1s >
aws ecs update-service --cluster qliktag-3x-cluster-production --service qliktag-service-api-production --force-new-deployment 1s >
aws ecs update-service --cluster qliktag-3x-cluster-production --service qliktag-service-redapi-production --force-new-deployment <1s >
echo "Successfully updated production cluster of container services"
```

## SNo #4



## SNo #5

Function name	Description	Runtime
<a href="#">qliktag-serverless-nodejs-production-adminapi</a>		Node.js 10.x
<a href="#">qliktag-serverless-nodejs-production-api</a>		Node.js 10.x
<a href="#">qliktag-serverless-nodejs-production-redapi</a>		Node.js 10.x

## SNo #6

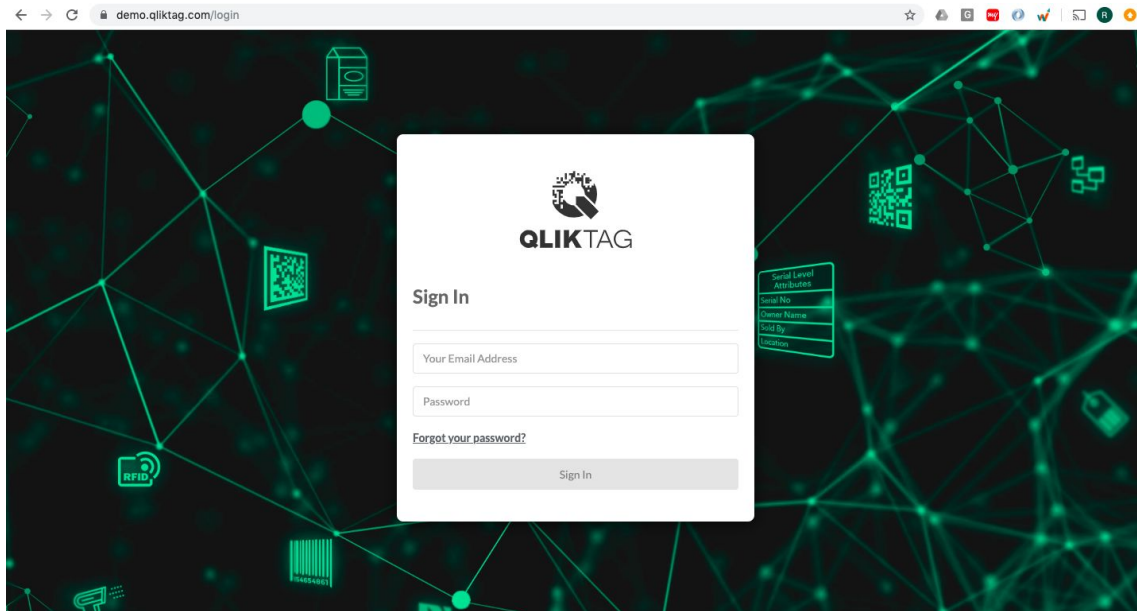
```
Angular CLI: 6.2.8
Node: 10.14.1
OS: darwin x64
Angular: 6.1.10
... animations, common, compiler, compiler-cli, core, forms
... http, language-service, platform-browser
... platform-browser-dynamic, router

Package                         Version
-----
@angular-devkit/architect       0.11.4
@angular-devkit/build-angular   0.11.4
@angular-devkit/build-optimizer 0.11.4
@angular-devkit/build-webpack   0.11.4
@angular-devkit/core             7.1.4
@angular-devkit/schematics       0.8.8
@angular/cdk                    6.4.7
@angular/cli                    6.2.8
@angular/flex-layout            6.0.0-beta.18
@angular/material               6.4.7
@ngtools/webpack                7.1.4
@schematics/angular              0.8.8
@schematics/update               0.8.8 (cli-only)
```

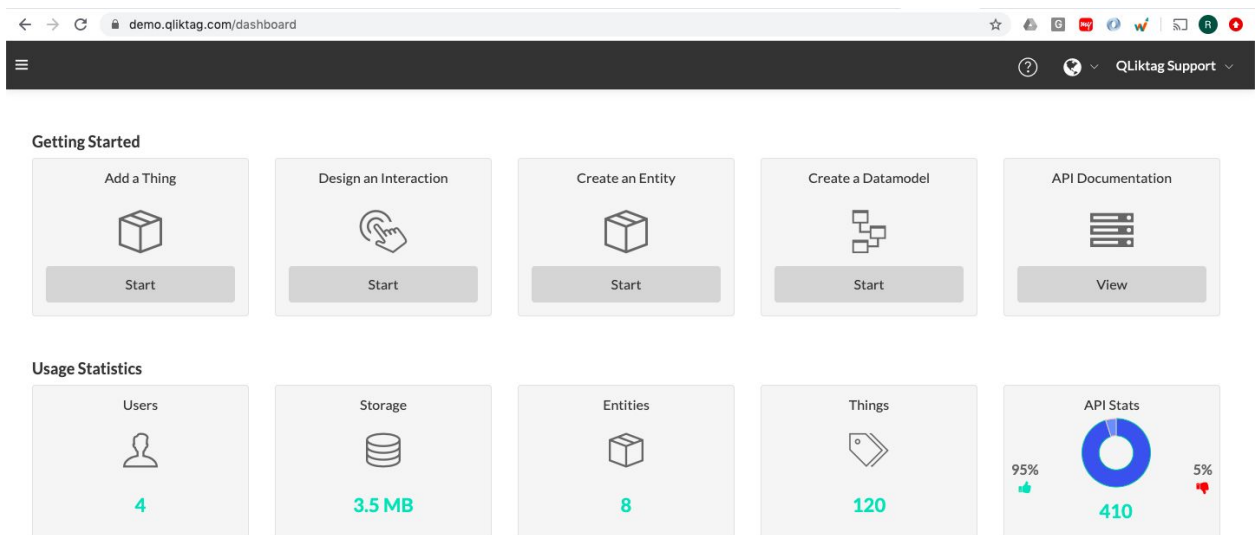
## SNo #7

```
demo.qliktag.com/api/health  
{  
  healthStatus: "green",  
  time: "2020-03-23T011:45:46.342Z"  
}
```

## SNo #8



## SNo #9



# SNo #10

demo.qliktag.com/users

Users + Create User

Users

First Name	Last Name	Email	Status	
neil	sequeira	neil.sequeira@qliktag.com	ACTIVE	
Demo	Qliktag	support@qliktag.com	ACTIVE	
Arjun	Viswanathan	arjun.viswanathan@qliktag.com	ACTIVE	
Dilip	Daswani	dilip.daswani@qliktag.com	ACTIVE	

4 total

## SNo #11

### GET API

GET /api/account

Get account details in controller

Parameters Cancel

Name	Description
<b>content-type</b> *required string (header)	Content-type of Response application/json
page integer (query)	Page number page - Page number
items_per_page integer (query)	Items per page items_per_page - Items pe
sort string(\$string) (query)	Sort by field Name sort - Sort by field Name
sort_order string (query)	Sort Order --
queryString string(\$string) (query)	&key1=val1&key2=val2 queryString - &key1=val1&

Execute Clear

Responses

Curl

```
curl -X GET "https://demo.qliktag.com/api/account" -H "accept: */*" -H "content-type: application/json" -H "Authorization: Bearer 25ff791f-bff4-479e-a2a1-3650a923f09c"
```

Request URL

```
https://demo.qliktag.com/api/account
```

Server response

Code Details

200 Response body

```
{
  "requestStatus": true,
  "errors": [],
  "warnings": [],
  "requestedAt": "2020-03-23T12:30:40.345Z",
  "data": {
    "account": [
      {
        "accountName": "Master Account",
        "status": "activated",
        "_accountId": "add121d0-ed88-11e9-b5b0-a1965f06ac07",
        "createdBy": {
          "user": ""
        }
      }
    ]
  }
}
```

Down



## POST API

**POST** /api/oauth

Create and Refresh the Authorization Token for validating the incoming request to the system

**Parameters** Cancel

No parameters

**Request body** **required**  -

Edit Value | Model

```
{
  "data": {
    "email": "#####qliktag.com",
    "password": "#####",
    "grantType": "password"
  }
}
```

Cancel Reset

Execute Clear

**Responses**

**curl**

```
curl -X POST "https://demo.qliktag.com/api/oauth" -H "accept: */*" -H
"Authorization: Bearer 25ff791f-bff4-479e-a2a1-3650a923f09c" -H "Content-Type:
application/json" -d "{\"data\":
```

**Request URL**

```
https://demo.qliktag.com/api/oauth
```

**Server response**

**Code** **Details**

200 **Response body**

```
{
  "requestStatus": true,
  "errors": [],
  "warnings": [],
  "requestedat": "2020-03-23T12:30:22.180Z",
  "data": {
    "userId": "4fe9be90-063a-11ea-8bd6-2183ec0290b1",
```

## SNo #12



## Welcome!

Mohit,

Demo is now using The Qliktag Platform – An IoT Platform for Everyday Consumer Products! The system Administrator Demo has invited you to join. In order to activate your user account, you will have to first set a secure password. Once you've set a password, you can login to the system using the password and details below:

## SNo #13

Through API

The screenshot displays an API testing tool interface for a POST request to `/api/assets/upload/file`. The tool is configured with the following parameters:

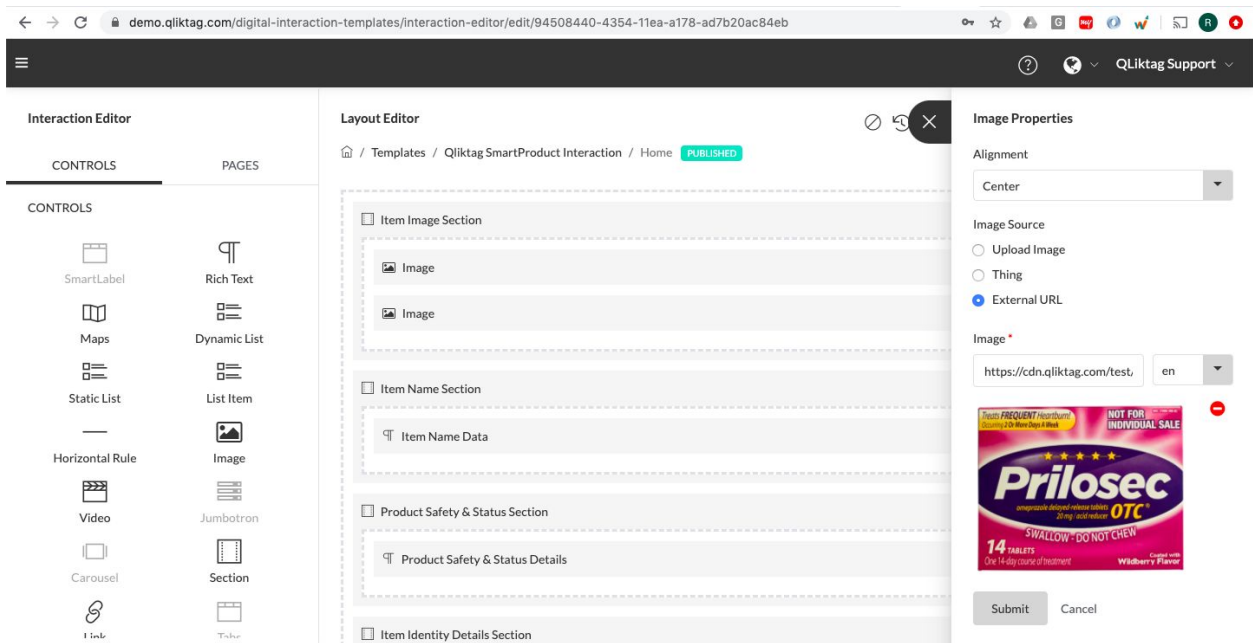
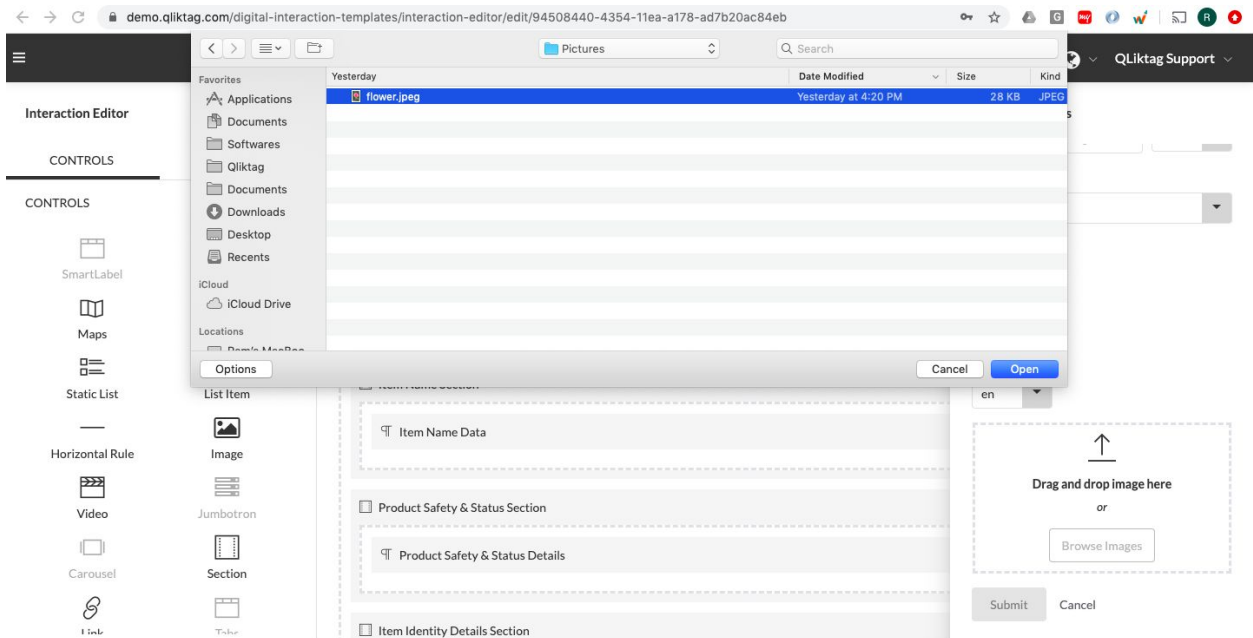
- `use_file_name`: true
- `unique_file`: true
- `file`: `roberto-nicko_unsplash.jpg`

The request body is set to `multipart-data`. The interface includes an `Execute` button and a `Clear` button. Below the request configuration, the `Responses` section shows the following details:

- `Request URL`: `https://demo.qliktag.com/api/assets/upload/file`
- `Server response Code`: 200
- `Response body`:

```
{
  "requestStatus": true,
  "errors": [],
  "warnings": [],
  "data": {
    "files": [
      {
        "filename": "406202_qliktag",
        "url": "https://cdn.qliktag.com/production/5da2edab97b667206158be15/roberto-nicko-unsplash-6gw_5d4298-unsplash.jpg"
      }
    ]
  }
}
```

# Through UI



## SNo #14

## SNo #15

#	Timestamp	RequestID	LogStream	DurationInMS	BilledDurationInMS	MemorySetInMB	Memo
1	2020-08-10T06:40:10.385Z	a7dfdcab-644b-441a-b7c6-adb1cb0a619a	2020/08/10[\$LATEST]e42bc96a718f42558dead47726443f4	187.33	200	512	495
2	2020-08-10T06:36:29.803Z	3c329011-1ef7-456e-b1b1-46e8431b6671	2020/08/10[\$LATEST]e42bc96a718f42558dead47726443f4	415.91	500	512	494
3	2020-08-10T06:36:28.385Z	38afdb74-9bc7-47e0-8c10-32a798b52adf	2020/08/10[\$LATEST]e42bc96a718f42558dead47726443f4	1881.79	1900	512	494
4	2020-08-10T06:36:27.081Z	8d2aad73-18a7-4fef-a335-121e88a5d823	2020/08/10[\$LATEST]1b2710c655ad4ad5a2e323793647cf5	410.99	500	512	472
5	2020-08-10T06:36:26.385Z	bb497249-6db6-4f8d-b7c9-00b8923afc48	2020/08/10[\$LATEST]e42bc96a718f42558dead47726443f4	339.74	400	512	462
6	2020-08-10T06:36:25.603Z	b1d255d9-eea4-4746-9a33-aa30699cc870	2020/08/10[\$LATEST]e42bc96a718f42558dead47726443f4	846.68	900	512	456
7	2020-08-10T06:36:25.340Z	ba382696-daaa-4753-ab70-0706a8a89f02	2020/08/10[\$LATEST]1b2710c655ad4ad5a2e323793647cf5	579.19	800	512	472
8	2020-08-10T06:36:25.256Z	4719b980-cd75-4b59-a745-1ceb7496bfb	2020/08/10[\$LATEST]c6394dbee8ae49e89c273bee5c7e45e9	498.91	500	512	478
9	2020-08-10T06:36:23.822Z	95417238-5d95-44af-a1af-6b2a5eb408bc	2020/08/10[\$LATEST]c6394dbee8ae49e89c273bee5c7e45e9	45.86	100	512	476

### Production Disaster Recovery (DR) Environment

S No	Review Item	Result (Pass/Fail)
1	Verify the database replica set connection	Pass
2	Verify the Elasticsearch connection	Pass
3	Verify the bitbucket pipeline status	Pass
4	Verify lambda service status in AWS console	Pass
5	Verify the lambda service running on NodeJS 10.x	Pass
6	Verify the web application is deployed with Angular correctly	Pass
7	Verify the API health status /api/health	Pass
8	Browse the web application and verify the page loading	Pass
9	Verify cloudwatch logs	Pass

## Sandbox Environment

Sandbox is a cloud account configuration with “<<controllerCode>>-sandbox” for enterprise customers in the same production release code. Only below verification items are applicable as it runs in the same production environment and verified under production environment deployment.

S No	Review Item	Result (Pass/Fail)
1	Browse the web application and verify the page loading	Pass
2	Verify the login to the web application	Pass
3	Navigate inside the web application and verify few pages	Pass
4	Test few API GET, POST operations	Pass
5	Verify emails send through application	Pass
6	Verify image uploads through API and web application	Pass
7	Verify few visual Interaction landing pages	Pass