DG DIGIT
Unit D.1

D01.01: DORIS+ Environments Manual

Consulting services on data analytics services applied to surveys and citizens feedbacks - Environment orchestration, automation, guidance and support on consultations

18/08/2020
Doc. Version: 1.0
Template Version: 2.5

Commission européenne, B-1049 Bruxelles / Europese Commissie, B-1049 Brussel - Belgium. Telephone: (32-2) 299 11 11.
Office: 05/45. Telephone: direct line (32-2) 2999659.

Commission européenne, L-2920 Luxembourg. Telephone: (352) 43 01-1.

This template is based on PM² v2.5

For the latest version of this template please visit the PM² Wiki

Document Control Information

Settings	Value
Document Title:	DORIS+ Environments Manual
Project Title:	Consulting services on data analytics services applied to surveys and citizens feedbacks - Environment orchestration, automation, guidance and support on consultations
Document Author:	Deloitte
Project Owner:	Roberto Barcellan
Project Manager:	Marc Vanderperren
Doc. Version:	1.0
Sensitivity:	Internal
Date:	24/08/2020

Document Approver(s) and Reviewer(s):

NOTE: All Approvers are required. Records of each approver must be maintained. All Reviewers in the list are considered required unless explicitly listed as Optional.

Name	Role	Action	Date
Blanca Martínez de Aragón	Project Manager	Review & approve	
Konstantinos Anastopoulos	Technical Reviewer	Review & Approve	

Document history:

The Document Author is authorized to make the following types of changes to the document without requiring that the document be re-approved:

- Editorial, formatting, and spelling
- Clarification

To request a change to this document, contact the Document Author or Owner.
Changes to this document are summarized in the following table in reverse chronological order (latest version first).

Revision	Date	Created by	Short Description of Changes
V0.1	18/08/2020	Deloitte	Initial version

Configuration Management: Document Location

The latest version of this controlled document is stored on Confluence.

Table of Contents

LIST OF FIGURES

No table of figures entries found. (still to be updated)

1. INTRODUCTION

1.1. Context

In its day-to-day work, the Commission is responsible for defining and implementing new policies, drawing up new legislation proposals, while conducting prior impact assessments, etc., in more than 30 different policy areas. In addition, the Commission also defines its internal strategy to grow as an organisation and to run on a daily basis.

As stated by the Communication on "Data, Information and Knowledge management", the challenges faced by the EU today require fast and effective solutions from the Commission. This results in the need for a modernization of the Commission's ways of working, and a strong need for strategic use of data, information and knowledge. In this context, DIGIT is prominent in some of the actions indicated in the work programme adopted by Information Management Steering Board (ISMB), the governance body resulting after the adoption of the Communication. In addition, the interim report of the IMSB has established a need to further develop and implement a business intelligence strategy and data analytics capabilities for the Commission during the next two years.

In this context, DIGIT has been working extensively in providing, among others, knowledge management and collaboration tools and methodologies, in the development of frameworks for data management and data interoperability, and in contributing to the enhancement of the data analytics capabilities of the Commission through execution of pilot projects and studies.

Open Public Consultations offer a means to the European citizens to provide their feedback on European regulation and the working of the Commission. Despite the large amount of opinions, needs and preferences expressed by citizens, governments' decision making processes are so far still not able to fully consume this unstructured and dispersed knowledge in order to extract meaningful knowledge and use it as input to decision making.

The DORIS aims to provide a more accurate analysis and a more tailored visualisation of the results of Open Public Consultations. The tool was initially developed by DG CNECT and handed over to DIGIT to be provided as a corporate service for the DGs of the Commission.

The generalization of the DORIS tool, developed by DG CNECT, was already performed by DIGIT in order to make it available to Member States' public administrations willing to process stakeholders consultations. Nowadays, DORIS can process results of surveys coming from EUSurvey and Better Regulation Portal. It allows users to analyze data from open and closed questions, and offers a dashboard through which users can visualize the results of the analysis.

At the current moment, after several years of providing services, the strong need for such a data analytics tool has been apparent, and the existing DORIS system has generated expectations between DGs. However, the current system has arrived to a limit in terms of scalability, operability and maintainability, and as a result of that, the redesign and new implementation of the system is required.

1.2. Objectives

The purpose of this document is to provide clear instructions on how to deploy DORIS+ in a new AWS environment.

2. STAKEHOLDERS

- The Directorate General for Informatics (DIGIT) as system supplier;
- Secretariat-General (SG) as the system owner;

3. PREREQUISITES

This chapter explains what is required in terms of environment, code, access and documentation before you can start a new DORIS+ deployment.

3.1. Set-up an EC2 environment

Those requirements can be met in two different ways, where method no 1 is recommended.

Method no 1 (recommended):

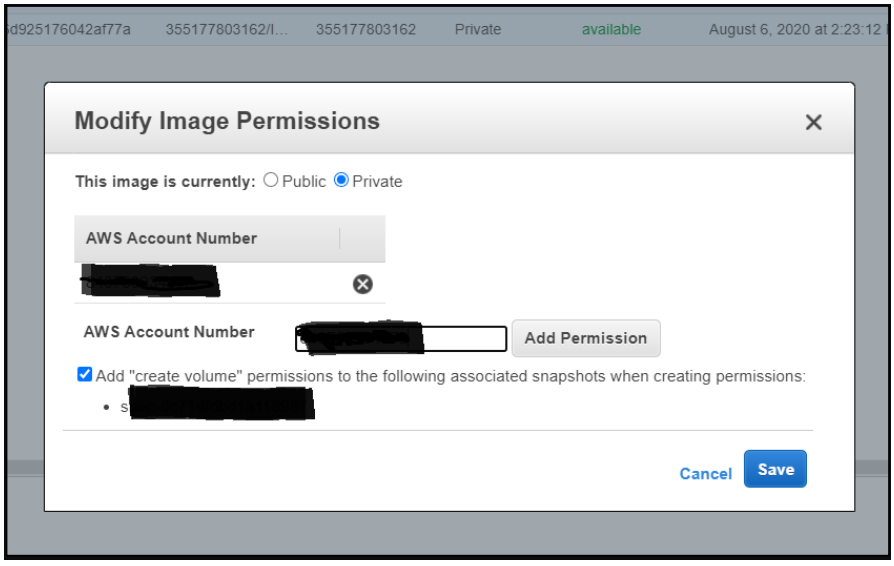
In this section, we'll be talking about how to configure an instance to be able to deploy the DORIS+ infrastructure. For convenience, an AMI has already been created named '**InfraInstance**' that can be used to create any future EC2 instances from which the Doris+ infrastructure and application can be deployed to any AWS account.

This method will be divided in two parts: In the first part, we will push the '**InfraInstance**' AMI from the master AWS account (CNECT-VICTORY) to the target AWS account where you want to deploy DORIS+. In the second part the launch of an EC2 with the AMI will be described.

Before you start, note down the target account number which can be found in *My Account* > *Account Settings* > *Account id*.

Part 1 - Push the AWS AMI to the target account:

- In the AWS console, log on to CNECT-VICTORY and verify whether you are in the Ireland region.
- Open the Amazon EC2 console, then click on *Image* > *AMI*
- Search on 'InfraInstance' in the "owned by me" space.
- Select the AMI '**InfraInstance**' in the list and then select *Actions* > *Modify Image Permissions*
- Fill in your target AWS account number and select the *Add "create volume" permissions* box as indicated below. Do **not** click save yet.
- Click on *Add Permission*
- Click on *Save*



If desired, futher documentation on the topic of AMI sharing can be found here.

Part 2 – In your target account, create an EC2 based on the pushed AMI

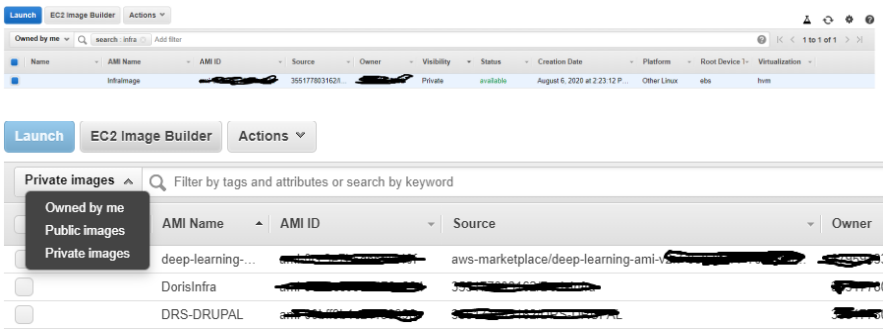
1. Open the AWS console for your target account where you want to deploy DORIS+
2. Create an IAM role for the EC2 and remember the name for later:

The role should contain the policies:

- AmazonEC2ContainerRegistryFullAccess
- AdministratorAccess policies.

1. Create the EC2

- In the AWS console, navigate to **EC2** => **AMI**
- Make sure the right region is set (Ireland).
- Click on *Owned by me* and select ***Private images*** instead. You will be able to see the shared AMI. If there are a lot of images, filter on 'InfraInstance'.



- Select the AMI, and click on *Launch*
- Change to "t2.small"
- Click *Next: Configure Instance Details*
- On *IAM role* add the created role in part 2.1
- Click *Review and Launch*
- Click *Launch*
- Create a new key pair and save it for later use (or use an existing key-pair)
- Click on *Launch instances*
- Click on *View instances* to track the progress
- Once the EC2 is created, save the IPv4 Public IP to be able to SSH to the instance

Once an an ec2 instance has been created using this AMI, all pre-requisites are subsequently considered met.

Method no 2 (not recommended):

Should there be a requirement for any reason to create a new EC2 instance from scratch, we will explain here how to do so. Note that the method below results in the same AMI as above, but this information is useful if at some point the AMI needs to be extended or upgraded.

1. Create an EC2 instance using the AWS console or the CLI.

- Select '**Red Hat Enterprise Linux 8 (HVM), SSD Volume Type**' as the operating system.
- As far as instance type, a **t2.small** would suffice but that can be scaled according to need.
- In terms of **VPC** and **subnet** there are no constraints.
- For storage we recommend a 30 GB EBS **General purpose SSD**
- Create a new key pair or use an existing one

2. After the instance is launched, here it what needs to be configured in order to be able to deploy the infrastructure successfully:

   - **AWS CLI**: Run the following command

- *sudo yum install zip unzip -y*
- *curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"*
- *unzip awscliv2.zipaws*
- *sudo ./aws/installsudo ./AWS/install*

- **Terraform**: Run the following command

  mkdir ~/bin
  curl https://releases.hashicorp.com/Terraform/0.12.24/Terraform_0.12.24_linux_amd64.zip
  unzip Terraform_0.12.24_linux_amd64.zip
  mv Terraform ~/bin

- **Docker CE edition**: Run the following command:

  sudo yum install docker

  sudo yum remove docker docker-common docker-selinux docker-engine

  sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

  sudo yum install docker-ce

- **Git:** Run the following command:

  yum install git

3.2. Set-up an S3 bucket with Talend installer dependency.

The current installer uses to deploy Talend can be found in the CNECT account in s3://drs-installers-repo/installers. The content needs to be downloaded locally and update to an s3 in the target AWS account.

Bucket names are unique per region, therefore the current name can not be used. The content of the installer needs to be added to the key s3://<BUCKET_NAME>/installers

This set up need to done only once per account, even though it contains multiple environments.

4. GIT REPO STRUCTURE

This chapter will explain what you need and what are the main components that will act in the deployment.

The Doris+ extension can be pulled via a git repository. In this repository, we will be focusing on the Terraform folder used for the deployment of the environment.

In this folder, you will have different subfolder: the modules, the commons, and three environments called Dev, Pre and Pro.

The modules folder contains all the component to be deployed by Terraform (bastion, batch, etc.), The commons folder contains the scripts used by Terraform to deploy specific resource as well as the python codes deployed by Terraform for the lambdas and the Docker image and the step-function template.

In the following tables, you will find the content of these folders in further details

- **Modules**

This folder contains the modules for each logical component of Doris plus Terraform resources deployed.

Name	Description
bastion	create bastion host
batch	create ecr repo, push docker image and create AWS batch - compute environment, job queue, job definition
documentdb	create documentdb cluster and secret manager
elasticsearch	create cognito pools, elasticsearch domain and configure nginx Kibana proxy.
orchestrator	create sqs, sns and step functions
s3	create s3 bucket and s3 endpoint
Talend	create windows instance and deploy code and Talend software components
vpc	create vpc, security groups and network acls
vpn	create AWS client vpn
APIGateway	Create 2 API gateways: Etranslate and s3 routing to Kibana
Lambda	Create all lambda functions required for the processing engine within Doris plus
lambda_config_file_update	Update the lambda configurations based on new parameters

- **Dev|Pre|Pro**

These folders are used to trigger Terraform and to configure the environment. Terraform configuration for each environment - Dev, Pre and Pro, by using the different value set up in vars.tf.

There are four files in each folder.

Name	Description

main.tf	Terraform file which calls all modules
provider.tf	AWS provider file
vars.tf	variables and values specific to each environment
versions.tf	Terraform version file

- **Commons**

These are the script used by Terraform to deploy specific resources. These script does not require to be manually triggered, it is part of the automated Terraform deployment explained bellow.

Name	Description
configure_nginx.sh	user data shell script to configure nginx instance
create_keypair.sh	helper shell script to create pem key for given env
create_ssc.sh	helper shell script to create a certificate for client vpn
deploy_Talend.ps1	userdata powershell script to deploy Talend
upload_win_installers.sh	helper shell script to upload Talend softwares to s3 bucket
nginx.conf	nginx configuration template file
localhost.rdp	preconfigured rdp file to logon onto a windows machine
deploy_lambda_to_s3.sh	user-data shell script to initialize the lambda deployment
redeploy_lambda_config.sh	user-data shell script to redeploy the lambda function with the updated configuration file
Code/ Doris-python-code	Python code for the Doris+ ETL
Code/ config_yaml.tpl	Template file to update the Doris+ python code config file
Stepfunction	Template file to update the step function with the new resources deployed via Terraform

- **Configs/variables**

Name	Description	Example
AWS_REGION	AWS region	eu-west-1
PREFIX	project prefix	DRS
ENV	environment	DEV
SUBNET	VPC CIDR : 10.{var.SUBNET}.0.0/ 22	3
BASTION_WHITELIST_CIDR	whitelist CIDR blocks	["0.0.0.0/0"]
NAT_EIP	if specify "new" will create EIP for nat else it will use specified PUBLIC_IP	34.168.3.45
INSTANCE_KEY_PATH	file path of key file	/keypair/drs-dev-instance
ACM_DIR_PATH	directory where vpn certificates are present	/acm
TALEND_INSTANCE	instance type	t3.xlarge

ETL_CODE_PATH	directory where etl code is available	/Users/pari/Music/digit-data-analytics-Doris-plus
PE_CODE_PATH	directory where process engine code is available	/Users/pari/Desktop/Doris-python-code
WIN_INSTALLER_PATH	directory where all windows installers are available	s3://drs-installers-repo/installers
WIN_INSTALLER_MAP	AWS region	eu-west-1
DOCDB_INSTANCE	map of instller file	
DOCDB_USER	documentdb master user	Doris
DOCDB_PASS	documentdb master user password	min 8 chars, must include special char, number and capital letter
NGINX_INSTANCE	instance type	t3.medium
ES_INSTANCE	instance type	r5.large.elasticsearch
EUSURVEY_ENDPOINT	api endpoint of eu survey	
EUSURVEY_USER	eusurvey api user	
EUSURVEY_PASS	eusurvey api password	Doris
BRP_ENDPOINT	api endpoint of brp	
BRP_USER	brp api user	/
BRP_PASS	brp api password	/

5. DEPLOYMENT INSTRUCTIONS

This chapter will provide detailed instructions to deploy DORIS+ in a new environment.

5.1. Grant full access to the user deploying

The user deploying the environment need have attached the following policies:

- AdministratorAccess
- AmazonEC2FullAccess
- AmazonEC2ContainerRegistryFullAccess
- AmazonVPCFullAccess
- ManageOwnCredentials
- AdminAccessIfMFA

Furthermore, access key credential for the user needs to be download.

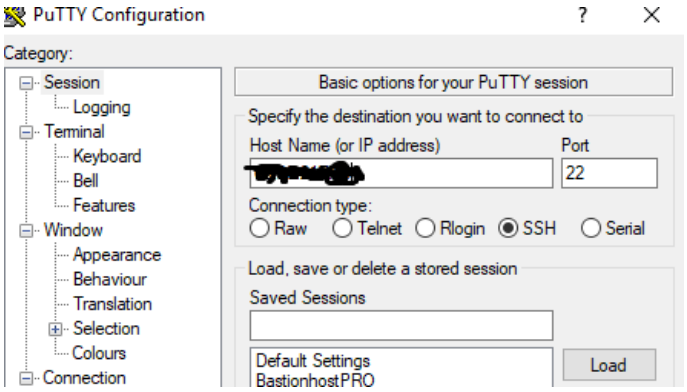
- Go to Identity and *Access Management (IAM) > Users*
- Click on the user that will deploy the environment
- Select *Security credentials >Create access keys*
- *Download the provided credentials*

5.2. Connect to the created EC2.

First, connect to the EC2 instance. There are multiple SSH clients so feel free to connect how you like. If you have your own way of connecting, you can go directly to part 2 of this chapter.

Part 1 - For Windows users, we will briefly explain how to connect with Putty.

- Download putty.exe here.
- Open putty.exe, and configure an ssh connection to the EC2. Add the IPv4 public IP from the EC2 you created and add port 22 (SSH)



Then in Connection > SSH > Auth > Authentification parameter add the private key for authentification created during the set-up of the AMI. You can then return to the session, save it and launch a connection by selecting open. The user to use by default is *****.

Furthermore, to access Talend to verify deployment, you can set up a tunnel via Connection > SSH > Tunnels then add 53389 as source port and localhost: 53389 as the destination, then click on add.

Commented [VP1]: To be looked at

Part 2

To finalise the EC2 configuration, type the following command:

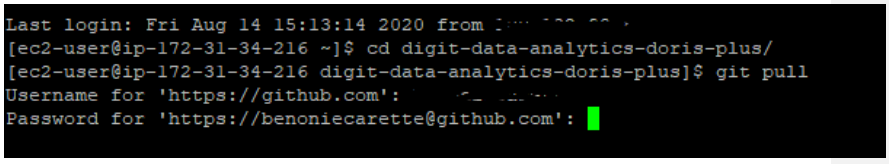
- *aws configure*
- Fill in the 'Access key ID' which you downloaded in part 5.1
- Fill in the 'Secret access key' which you downloaded in part 5.1
- Press enter to remain in the region or change if not correct (eu-west-1)
- Press enter to keep the default output format (JSON)

5.3. Pull/clone the repository.

Now that we connected to the EC2, we need to make sure that we have the up to date version of the git repository for Doris+

If you used method no 1 to launch the EC2, the git repository will be present, you then need to pull the latest version of the code. To do so, you can type the following command:

- *cd /home/ec2-user/digit-data-analytics-doris-plus/*
- *git pull*



If you used method 2 in part 4, the git repository will not be present, you then need to clone the latest version of the code. To do so, you can type the following command:

- *git clone https://github.com/ec-europa/digit-data-analytics-doris-plus.git*
- *cd digit-data-analytics-doris-plus/*

5.4. Run Terraform deployment

5.4.1. Set-up account-wide configuration

In order to be able to run terraform some configuration needs to be applied account wide. Execute this step only once per account. In order to apply this configuration run

sh ../commons/configure_account.sh

In a first step this script will create two account wide IAM roles for ElasticSearch. In a second step it will ensure that the terraform state is stored in a S3 bucket. It will create the S3 bucket and adapt terraform backend configuration.

The script will check for all the resources if they exist and if not create them. In case they already exist it will display a warning but continue configuration.

5.4.2. Deploy the Environment

The deployment of Terraform is streamlined, to do so, you will need to first create a key pair used by EC2 to connect, create a certificate for the VPN resource, ensure that the variable of the environment to be deployed is up-to-date; init Terraform and finally deploy.

First, you need to go to the required environment by running the following command.

- Make sure you are in the */home/ec2-user/digit-data-analytics-doris-plus* folder:
- cd terraform/<your environment to deploy>

the environment to deploy can be dev, pre or pro. Afterwards, you can follow these 4 steps.

step-1 : generate keypair:

To generate the keypair, type the following command and then press enter without providing a passphrase. This means you have to put in the command and press enter thrice (command + twice for empty passphrase)

sh ../commons/create_keypair.sh <dev_pre_pro>

This step is not included in the Terraform code itself to ensure that keys don't have to rotate with each terraform apply – destroy cycle

step-2 : generate a certificate for client vpn

Similarly run the certificate create for the VPN resource

sh ../commons/create_ssc.sh <dev_pre_pro>

step-3: Modify vars.tf file in accordance with a given environment

By default, the variable will be set-up to run the environment without issue. However, there is two key variables to check to ensure a clean deployment:

1. "AWS_REGION" need to be set up to the application region (in this case: eu-west-1")
2. "NAT_EIP" is set by default to new, and therefore will create a new EIP. However, Talend is connecting to BRP that require to whitelist an IP. Therefore, you are required to create an elastic IP and to request a whitelisting to BRP. When the IP is created and whitelisted, you can replace "new" with the new IP value. Do not remove the double quotes.
3. "WIN_INSTALLER_PATH" need to be set to to s3://<BUCKET_NAME>/installers where BUCKET_NAME correspond to the s3 environment set up in section 3.2

To modify these variable, from the environment folder run the following command:

vi vars.tf

From the vim interface, press “i” for insert, modify the variable then press escape followed by “*:wq*” to write down the change.

Step-4: Run Terraform commands

Finally to deploy the environment, go to the environment you want to deploy; init Terraform plan your deployment and finally deploy the resources by using the following command.

- *cd /home/ec2-user/digit-data-analytics-doris-plus/terraform/<your_environment>*
- *terraform init --backend-config=config.tfbackend*
- *terraform plan* (the ‘apply’ also performs a plan operation, so this one can be skipped)
- *terraform apply*
- When Terraform requests confirmation to apply actions, type *yes*

The deployment will fail if a resource has an identical name as the one created if it is the case, delete the resource and re-run Terraform apply.

After the deployment, Terraform will generate an output as the picture below. Save the value, as it contains key value to connect to the different resources.

The output is the following:

Name	Description
bastion	Ip and command line to ssh into the bastion
Client_vpn	Key-value to connect to the VPN
Documentdb/endpoint	Cluster id of the documentDB
Elasticsearch/IAM_arn_for_user_group	ARN of user role used for fine grain access in Kibana
Elasticsearch/es_name	Reference of the elastic search cluster
Elasticsearch/es_name/Kibana_public_url	Public url used to connect to Kibana without required tunnelling in the VPC
Elasticsearch/es_name/Kibana_vpc_url	Private url used to connect to Kibana with required tunnelling in the VPC
User_pool	User pool name
S3/bucket_name	The bucket containing Talend code
Talend/password	The password to connect the Talend windows VM

Talend/ssm_cmd	Command-line to connect to Talend windows VM via tunnelling
nginx	Private and public ip of nginx

To retrieve the created output, run the following command:

- *terraform output -json > outputs.json*

All Talend jobs and step functions will be automatically scheduled in your environment to run daily and Doris+ is now fully working. However a couple extra steps are required to allow further administration of Doris+.

5.5. Post-Installation steps

After the installation execute the following script:

sh ../commons/finalize_environment.sh <dev_pre_pro>

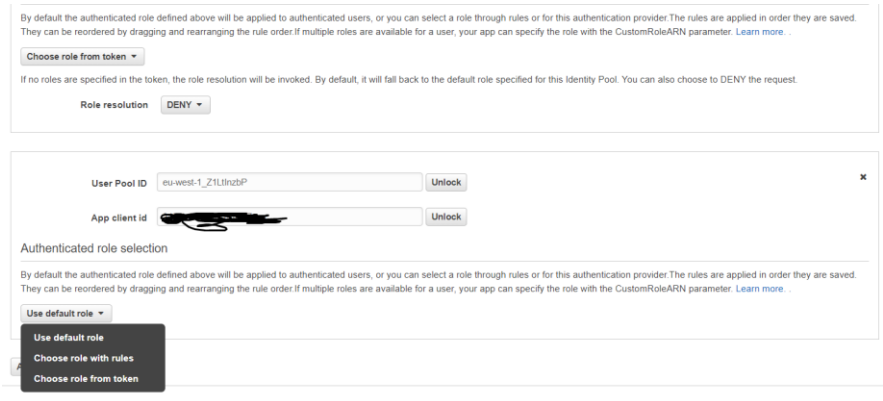
This script will ensure the correct indices are created in elastic search.

5.6. Configure Cognito

Most of Cognito deployment is automated via Terraform, but because of limitations 2 more steps need to be done manually in Cognito.

5.6.1. Modify the Authenticated role selection

- Connect to the AWS cognito service via the AWS management console
- Select *Manage Identity Pools*
- Go to *DRS_<created environment>_IDENTITY_POOL*, where *<created environment>* corresponds to the environment you just deployed.
- In the identity pool, select *Edit identity pool* on the upper right side of the screen.
- Finally go to *Authentication providers*



- The *authenticated role selection* role is currently select as *Use default role* and need to be changed as *choose role from token* and then *Role resolution* is required to be set as DENY as it is deployed on the first application. Do this only for the top one.
- Click on *Save Changes* when you are done.

5.6.2. Create a user and add it to the master user group

This paragraph describes how to create new users and how to provide them the access rights by adding them to a user group.

New user accounts need to be created by a DORIS+ admin user, for which the administrator will require the email address of the new user and the DG the user belongs to.

You can do this by running

sh ../commons/add_cognito_user.sh <dev_pre_pro> <emailadress>

When the user is created, add it to the master user group. navigate to the DORIS+ user pool > users and groups > groups

Select the group DRS-<ENV>-ElasticSearchMasterUser , then click on *Add users* and add the newly created user.

5.7. Set up of Kibana

When the master user is created, you can then connect to Kibana via the Kibana public URL generated by Terraform (as described in the output). You can use the user you created and password.

5.7.1. Upload the Kibana Dashboard

In the folder /home/ec2-user/digit-data-analytics-doris-plus/Kibana_dashboard you will find a ndjson file contains the Kibana dashboard to upload.

- In Kibana toolbar, navigate to Settings > Saved Objects.
- Click the Import button and select the dashboard file.

5.7.2. Creating and mapping Open Distro Security Roles

For each DG user group, a specific Open Distro Security Role needs to be created and mapped to the corresponding IAM role. The IAM role ARN can be found in the Terraform output.

New Open Distro Security Roles can be created and mapped to IAM roles using the Kibana UI or the _opendistro/_security operations in the REST API. Below follows a walk-through on how to create a new Open Distro Security Role for a DG user group and how to map it to the IAM role for the DG user group from the Kibana UI.

More information on creating Open Distro Security Roles and which permissions can be configured can be found on the following link. https://opendistro.github.io/for-elasticsearch-docs/docs/security-access-control/users-roles/#create-roles.

A description about the default action groups that can be used to define the permissions for the new roles can be found on the following link. https://opendistro.github.io/for-elasticsearch-docs/docs/security-access-control/default-action-groups/

5.7.2.1. Walkthrough for creating an Open Distro Security Role:

1. Log in to Kibana as a master user (or any other user with permissions to configure security settings)
2. Go to the security tab and select roles. In the roles tab you can create a new role from scratch or copy an existing role, which you can afterwards adapt. This walkthrough will

describe how to define the proposed roles for DG level access, specifically for the Secretariat General, starting from a copy of the default Kibana_user role.

3. Click the "copy" button for the Kibana_user role. A copy of the role definition is generated. You can now configure the permissions for the role you want to create. The following permissions need to be configured for the example DG-level access role for the Secretariat General:
    a. In the overview tab: adapt the role name to "KibanaUser_DG-level-Access_SecretariatGeneral"
    b. Advance to the cluster permissions tab where the "cluster_composite_ops" action group should be selected
    c. Advance to the index permissions tab and add additional index permission to the standard Kibana_user index permission to provide access to the OPC's from the Secretariat General
        i. Add an index pattern "feedbacks"
        ii. Add action group(s) to define the actions the user can perform on the index and select read
        iii. Add a document-level security query to restrict access to the OPC's of their own DG:

```
{
        "match":{
                "consultation_units": " SECRETARIAT-GENERAL"
        }
}
```

        iv. Note that the consultation_units field is inconsistent; all values in the "consultation_units" field that can be visible to the user group should be added to the document level security query.

    d. Advance to tenant permissions and add the Kibana_all_read action group to the global permission to make sure end-users cannot adapt or remove the visualisations and dashboards on the global tenant.

4. Confirm creation of the new role

5.7.2.2. Walkthrough for mapping an Open Distro Security Role to IAM role:

1. Log in to Kibana as a master user (or user with permissions to configure security settings)
2. Go to the security tab and select role mappings
3. In case the Open Distro Security Role is already in the list, select edit role. Otherwise, click create new role mapping and select the desired Open Distro Security Role from the list.
4. Add the ARN of the AWS IAM role to the list of backend roles.
5. Submit mapping

Note that the tab "hosts", which maps Open Distro Security Roles to hostnames and IP addresses, is not available because the Elasticsearch domain is located inside a VPC.

5.8. (Optional) Subscribe to the SNS topic to receive ETL status email

- Log in to AWS management console and navigate to the Amazon SNS > Subscriptions

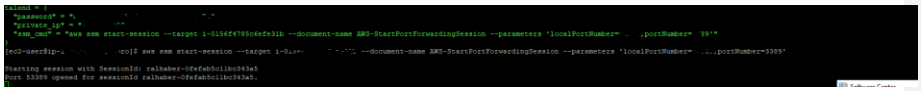
- Select Create subscription
- The subscription is called DRS-<ENV>-SNS-TOPIC, Select protocols, then finaly email.
- Add your email and save.

6. VALIDATION

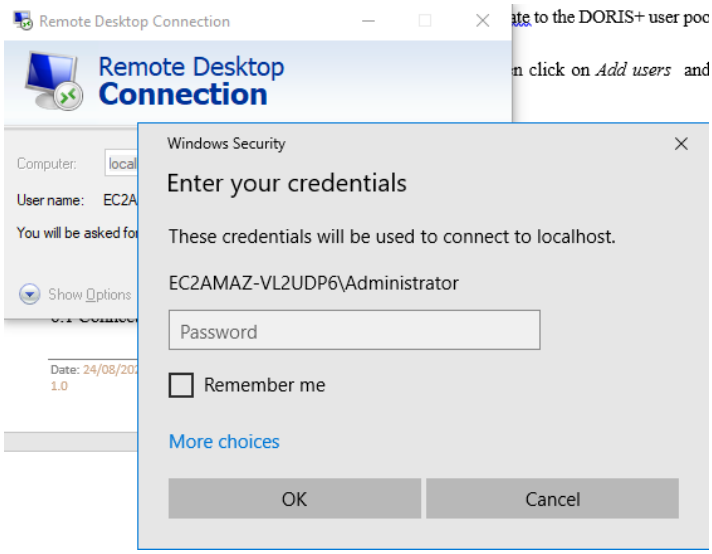
6.1. Talend

6.1.1. Connect to Talend

To connect to Talend you need to use the command generated in the output Talend/ssm_cmd
You need to have set up the tunnelling as described in part 4.



In your local windows, go to Remote Destock connection, connect to localport:53389. Then add the password generated by the output. (username: Administrator)



You can then connect to Robot3 to connect to the documentDB cluster and check it received data

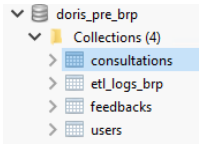
6.1.2. Set up Robot3T

Cfr. Everis Talend documentation

6.1.3. Results Validation

In order to verify that the extraction completed correctly, it is possible to check the extracted data in MongoDB from Robot 3T.

In the database with the name provided in the configuration file, four collections should have been created:



Consultations collection

In the consultations collection, after executing the following query, the loaded consultation can be seen, whose structure and content should look as follow:

```
db.getCollection('consultations').find({"consultationId" : "7608006"})
```

Feedbacks collection

Moreover, after executing the following query, the feedbacks collection should return 7 documents[1]:

```
db.getCollection('feedbacks').find({"consultationId" : "7608006"})
```

[1] Please note that this number could change because this consultation is still open, therefore new feedbacks could be added after that this document has been written.

Key	Value
(1) ObjectId("5e7a0aeaa975aa0efc13727e")	{ 15 fields }
(2) ObjectId("5e7a0aeaa975aa0efc137280")	{ 14 fields }
(3) ObjectId("5e7a0aeaa975aa0efc137281")	{ 14 fields }
(4) ObjectId("5e7a0aeaa975aa0efc137282")	{ 15 fields }
(5) ObjectId("5e7a0af6a975aa0efc13728a")	{ 14 fields }
(6) ObjectId("5e7a0af6a975aa0efc13728b")	{ 14 fields }
(7) ObjectId("5e7a0af6a975aa0efc137287")	{ 15 fields }

Each of them will have a structure similar to this one.


```
{
    "_id" : ObjectId("5e8738290996650890d56ea4"),
    "consultationId" : "7608006",
    "alias" : "Ares(2020)1505101",
    "language" : "DE",
    "feedbackId" : "510028",
    "questionType" : "FREE_TEXT",
    "country" : "DEU",
    "groupId" : "e4b2de28f804d4e8865ce6c73a29ee56",
    "questionId" : "e4b2de28f804d4e8865ce6c73a29ee56",
    "answerText" : "Zu dem bisherigen Ziel der EG-Abfallverbringun
    "createdDate" : ISODate("2020-03-31T13:40:32.000Z"),
    "lastModifiedDate" : ISODate("2020-03-31T13:40:37.000Z"),
    "user" : {
        "name" : "Ulrike Bönisch",
        "userType" : "PUBLIC_AUTHORITY"
    },
    "kind" : "BRP"
}
```


Inside the same feedbacks collection, one document will also represent an attachment uploaded by the user, whose structure should look as follows (usually it's the last one in the collection):


```
/* 7 */
{
    "_id" : ObjectId("5e7a0af6a975aa0efc137287"),
    "consultationId" : "7608006",
    "language" : "EN",
    "feedbackId" : "509070",
    "questionType" : "Free Text",
    "country" : "CHE",
    "groupId" : "e4b2de28f804d4e8865ce6c73a29ee56",
    "questionId" : "e4b2de28f804d4e8865ce6c73a29ee56",
    "answerText" : "Many low- and middle-income countries do not have the capacity to safel
    "createdDate" : ISODate("2020-03-16T20:37:39.000Z"),
    "lastModifiedDate" : ISODate("2020-03-16T20:41:04.000Z"),
    "uploadedAt" : ISODate("2020-03-24T00:00:00.000Z"),
    "user" : {
        "name" : "Kupka Rachael",
        "userType" : "NGO"
    },
    "kind" : "BRP",
    "attachment" : {
        "fileType" : "pdf",
        "extractedFilePath" : "attachments/Ares(2020)1505101/ExtractedFiles/509070",
        "originalFilePath" : "attachments/Ares(2020)1505101/UploadedFiles/509070",
        "fileHash" : "8149381b72b542eff55cf3b1a88651ce",
        "totalChars" : 1757,
        "fileBytes" : 119331.0,
        "name" : "090166e5cd264685.pdf",
        "originalText" : "Lead and ULAB Recycling References: 1 Ericson, B., Landrigan, P.,
    }
}
```


Users collection

The Users collection contains the information about the users that answered the consultations: each row corresponds to one unique user and they will look as follows:

```
/* 10 */
{
    "_id" : ObjectId("5e6f97017b419f35049f888a"),
    "userId" : "96592dd1211c58fcfa248281a07874b0",
    "name" : "Henk Jan Nix",
    "country" : "NLD",
    "consultationId" : "7608006",
    "feedbackId" : "508385",
    "userType" : "BUSINESS_ASSOCIATION"
}

/* 11 */
{
    "_id" : ObjectId("5e6f98fc7b419f35049f8a64"),
    "userId" : "804803df7a4f15c045e57b313d68acc0",
    "name" : "Emma ACHILLI",
    "country" : "IRL",
    "consultationId" : "7592309",
    "feedbackId" : "508437",
    "userType" : "NGO"
}

/* 12 */
{
    "_id" : ObjectId("5e6f98fc7b419f35049f8a65"),
    "userId" : "311f6ea8b7805be3ed87c417e41ca62e",
    "name" : "Béatrice GOREZ",
    "country" : "BEL",
    "consultationId" : "7592309",
    "feedbackId" : "508261",
    "userType" : "NGO"
}
```


Etl_logs collection

The etl_logs collections for BRP contain all the event logs of the job. The contained document will have the same structure and meaning of the logs described previously for the EUSurvey ETL Job.

6.2. Run Data load step function

To do so:

- Log in to AWS management console and navigate to the AWS step-function service > DRS-<ENV>-DATA-LOAD-STATE-MACHINE> Start execution.
- Select a run for one consultation by adding the following JSON to the execution:

```
{
  "consultations": [
    "251460",
    "289427"
  ]
}
```

Wait until you have a full run.

6.3. Connect to Kibana dashboard

If the data has been fully loaded, the data will appear in the dashboard

7. COMMON ERRORS

This chapter bundles common errors as well as how to solve them. This chapter can be extended over time.

7.1. Common errors during deployment

7.1.1. The resource already exist

Sometimes Terraform might try to deploy resource that matches the exact name of the resource created manually in the past. As described in the picture

```
module.vpn.null_resource.apply_security_group: Creation complete after 4s [id=1256051499
15434633]

Error: Error creating IAM Role DRS-PRE-ETL-INSTANCE-ROLE: EntityAlreadyExists: Role with
name DRS-PRE-ETL-INSTANCE-ROLE already exists.
        status code: 409, request id: 93665d80-7be5-4bc0-8e35-a2f7591ef5da
```

In this case, Terraform can destroy the only resource it deployed beforehand. Therefore you will need to delete the resource manually.

7.1.2. Someone made a manual modification to the environment it already deploy.

Terraform need to be refresh with the current status of the environment. You can run the command

Terraform refresh

7.1.3. No more room for further elastic IP or VPC.

If the elastic IP variable is set to new it will generate a new EIP. However, the number of EIP is limited by default in the AWS account. You, therefore, need to delete one EIP that is not in use or add an existing one in the deployment. You can also remove this limit.

7.1.4. Error with the installer files

"Error: Error running command 'bash -e ../commons/upload_win_installers.sh s3://drs-installers-repo-dev/installers /home/ec2-user/digit-data-analytics-doris-plus drs-pre-data-846759934673 AW SCLI64PY3.msi,jre-8u251-windows-x64.exe,robo3t-1.3.1-windows-x86_64-7419c406.zip,TOS_BD-20200219_1130-V7.3.1.zip': exit status 18. Output: red 0%)"

This might mean the location of the installers is wrong or the files are corrupted. However this can also be because the files got corrupted during the process itself. Therefore, before you try to upload the files again to the bucket, run terraform apply again to give it another try.

8. ANNEX – GLOSSARY

The following terms are regularly used thought this report:

Table 1 Glossary

Term	Definition
Docker	Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels. All containers are run by a single operating system kernel and therefore use fewer resources than virtual machines.

Terraform	Terraform is an open-source infrastructure as code, software tool created by HashiCorp. It enables users to define and provision data centre infrastructure using a declarative configuration language known as HashiCorp Configuration Language, or optionally JSON.