



Hack-A-Tech Track

Hack-a-Tech

1st Party Data Tool Tightlock: Activate your 1st Party Data Simple and Secure
20th of June

Hack-a-Tech

What is the Hack-a-Tech Series?

- Hands on tech sessions
- Designed by and for engineers & data scientists
- For EMEA partners
- Innovative & Scalable solutions

Introduce yourself in the chat

Name, Company, Function and Summer holiday destination



Sarai
Privacy Solution
Engineer



Katerina
Customer Solution
Engineer



Floris
Partner Engagement
Manager

Hack-a-Tech Breakouts Team



Kuba



Alberto



Inbar



Miguel



Jakub



Max



Kevin



Ehsaan



Caio



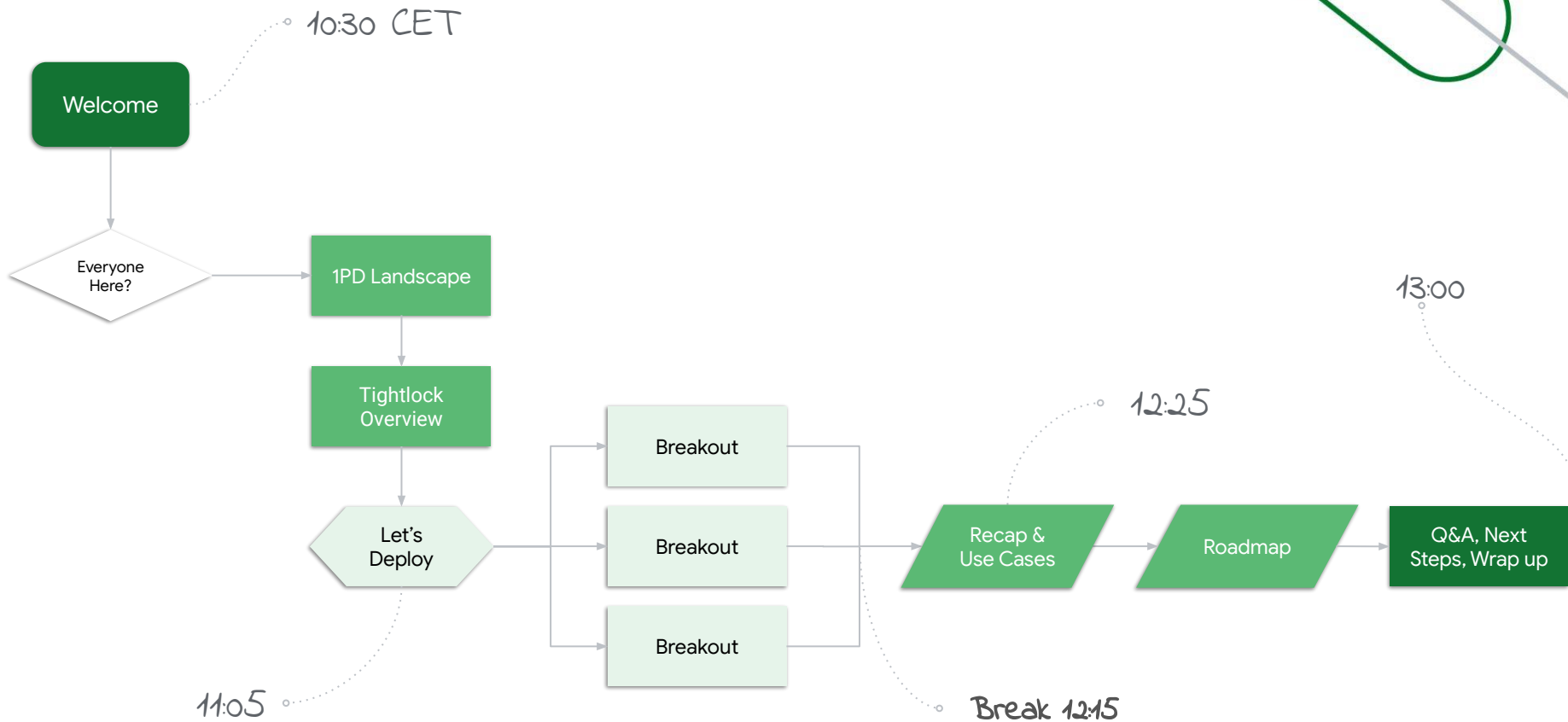
Tom



Niels

Breakout Leads

Today's Agenda



Housekeeping



Please double check you are on mute and
please keep camera on if possible



If you have questions, please use chat



Have Fun



Hack-A-Tech Track

1PD Landscape

Katerina



What is First-Party Data?

Why is 1PD important?

Why are they important?



Users get **personalization** as a
“standard of service”

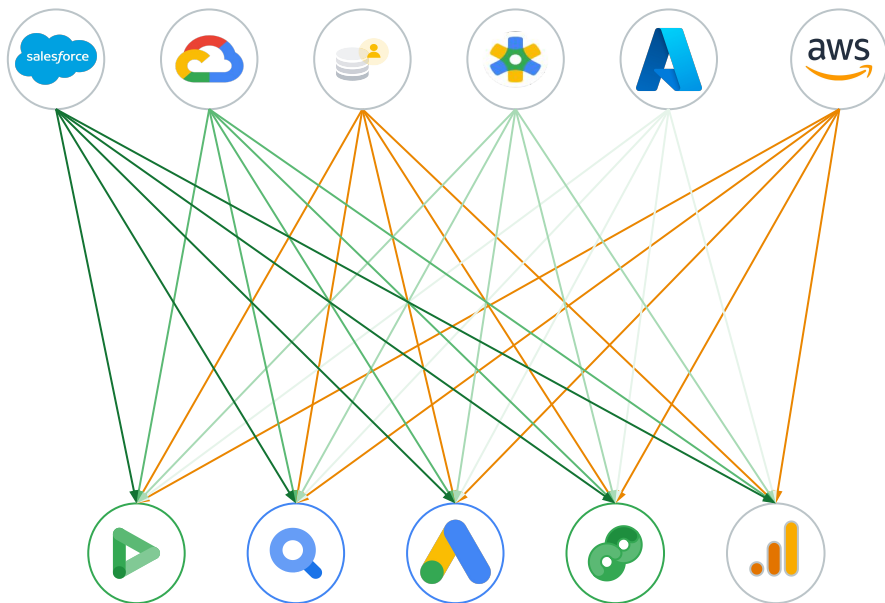


Reach new consumers effectively &
increase retargeting capabilities



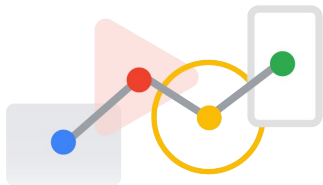
2x incremental revenue from a single
ad placement & **1.5x improved cost**
efficiency

Challenges of First-Party Data

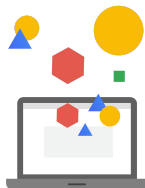


Challenges of First-Party Data

Data Challenges



Process Challenges



Clear Needs



First-Party Data Landscape

Two steps to detangle the web of 1PD:

1

Simplicity

Helping clients identify their first party data, and easily connecting it to the relevant Google advertising features is key.

2

Automation

Automation ensures activating first-party data is still relevant and timely to recent impressions or activities.



Hack-A-Tech Track

Tightlock Overview

Sarai

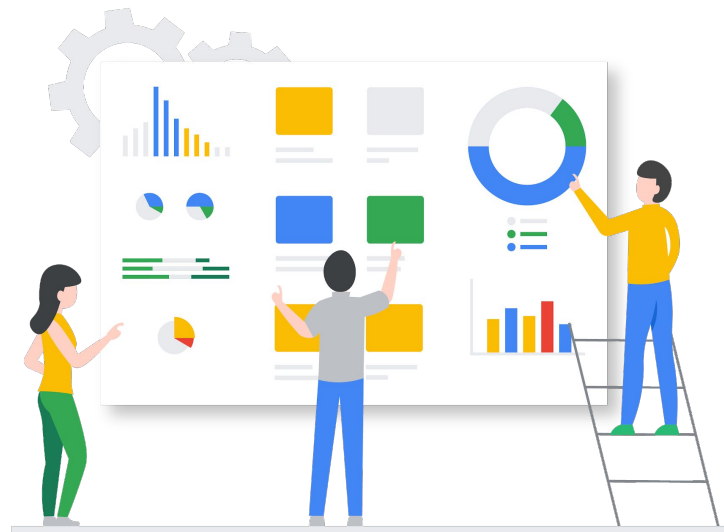
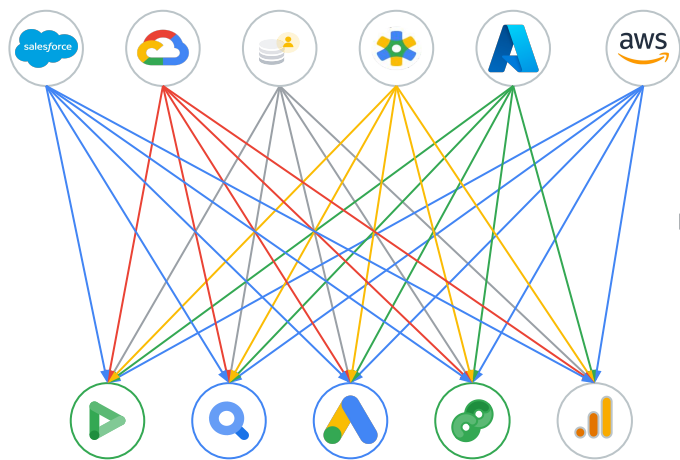


The Solution - Tightlock

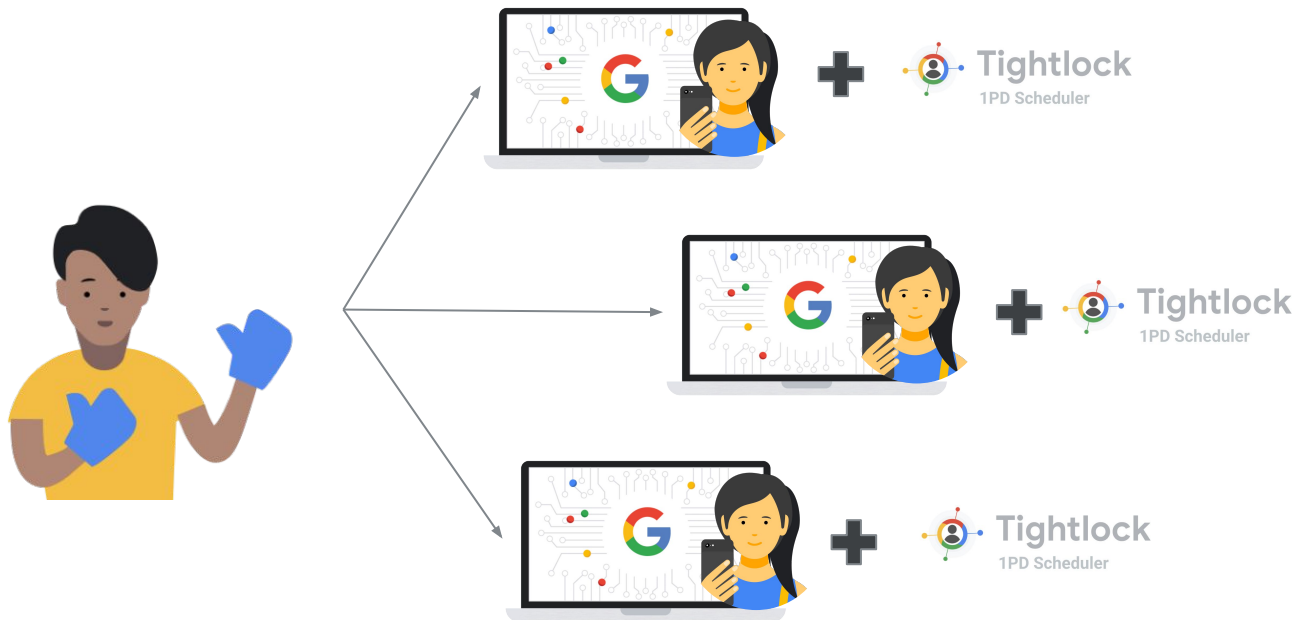


Tightlock
1PD Scheduler

Tightlock - Reduce 1PD Complexity



Tightlock - A Workspace for Each Client



What Do Clients Need to Start Using Tightlock



Collecting and preparing the first party data to send is important. Then the client also needs to make sure they have the credentials and access rights to the resources and destinations they would like to use.

First Party Data



Step 1

Credentials



Step 2

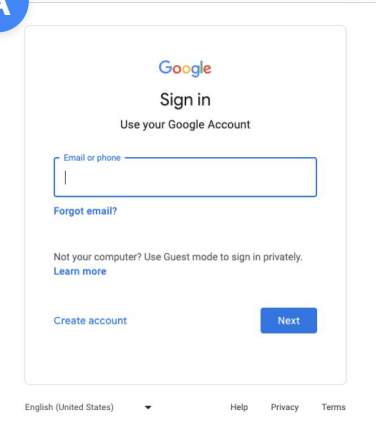
Access Rights



Step 3

First, advertisers sign up, create a workspace, and manage users

A



Google
Sign in
Use your Google Account

Email or phone

[Forgot email?](#)

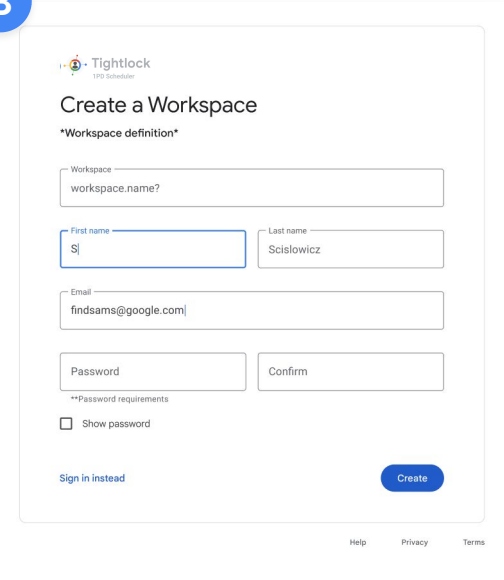
Not your computer? Use Guest mode to sign in privately.
[Learn more](#)

Create account [Next](#)

English (United States) Help Privacy Terms

Sign into a Google account

B



Tightlock
TPO Scheduler

Create a Workspace

Workspace definition

Workspace
workspace.name?

First name Last name
Sj Scislowicz

Email
findsams@google.com

Password Confirm

**Password requirements

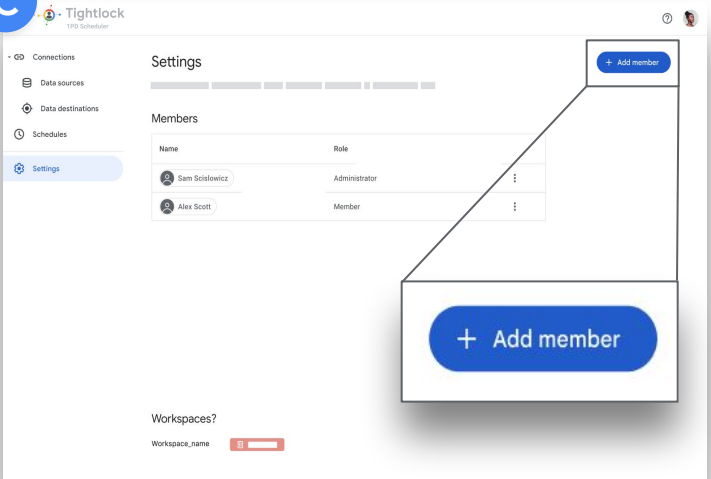
☐ Show password

[Sign in instead](#) [Create](#)

Help Privacy Terms

Create a workspace

C



Tightlock
TPO Scheduler

- Connections
- Data sources
- Data destinations
- Schedules
- Settings**

Settings

[+ Add member](#)

Members

Name	Role
Sam Scislowicz	Administrator
Alex Scott	Member

[+ Add member](#)

Workspaces?

Workspace_name

Manage members / users

The next step is do create connections and add resources

D

Tightlock
IPO Scheduler

Connection wizard

1 Select data source Summary if needed 2 Select use case Summary if needed 3 Configure Summary if needed 4 Select destination Summary if needed 5 Set a schedule Summary if needed

Previous Next

What will you use this connection for?
The use case you choose will determine the destination table's schema. You can preview the destination fields in the next step.

- Customer segmentation**
Measure campaign performance sliced by imported customer segments
- ROI calculation**
Measure return on investment by importing transaction or conversion data
- In-app performance**
Analyze campaigns by importing in-app measurement data, such as installations, activations, in-app purchases, or retention
- Cross-platform analysis**
Calculate reach and frequency or measure attribution among platforms by importing touchpoints and stitching them with Google ad data
- Custom modeling**
Create custom models to predict measurements (such as lifetime value) by importing transactions, customer segments, or other touchpoints as features
- Campaign comparison**
Compare Google ad campaign performance with other campaigns by importing campaign and cost data

E

Tightlock
IPO Scheduler

Sources

+ Add data source

- Connections
- Sources
- Account / Settings?

Name	Location
Data A1	Data A1
Data A2	Data A2
Data A3	Data A3
Data A4	Data A4
Data A5	Data A5
Data A6	Data A6
Data A7	Data A7
Data A8	Data A8
Data A9	Data A9
Data A10	Data A10

...and they then can see it in the resources page

Clients create connections using a step-by-step wizard

Tightlock - Features



Open Sourced

Cloud Agnostic

Cost Efficient

Extensible

Maintained

Secure

Tightlock - Features



Open Sourced

Cloud Agnostic

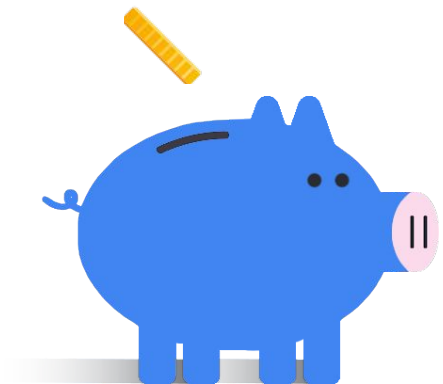
Cost Efficient

Extensible

Maintained

Secure

Tightlock - Features



Open Sourced

Cloud Agnostic

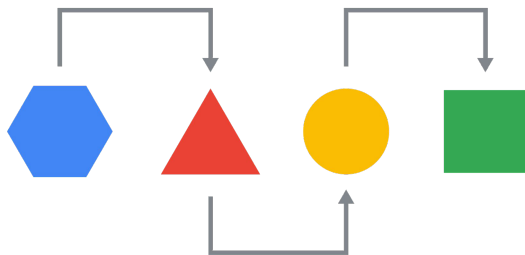
Cost Efficient

Extensible

Maintained

Secure

Tightlock - Features



Open Sourced

Cloud Agnostic

Cost Efficient

Extensible

Maintained

Secure

Tightlock - Features



Open Sourced

Cloud Agnostic

Cost Efficient

Extensible

Maintained

Secure

Tightlock - Features



Open Sourced

Cloud Agnostic

Cost Efficient

Extensible

Maintained

Secure

Tightlock - Summary



Tightlock
1PD Scheduler

Open Sourced

Cloud Agnostic

Cost Efficient

Extensible

Maintained

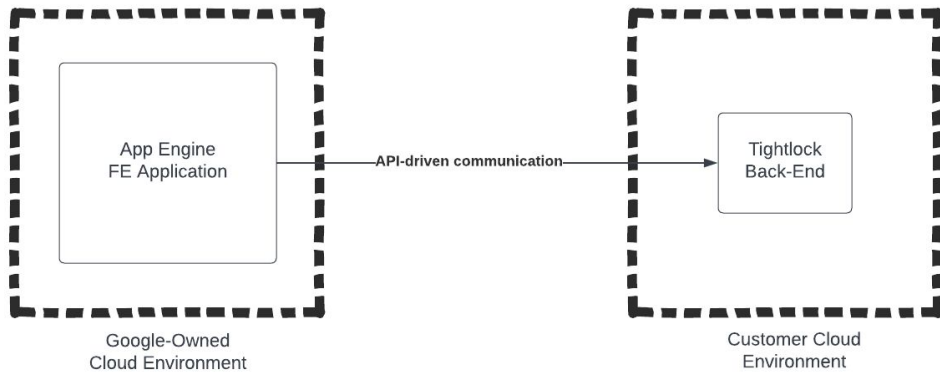
Secure

Privacy & Measurement

Tightlock Architecture

Tightlock - How it Works

API-Driven Backend



```
tightlock_api > app > main.py
35 @v1.post("/connect")
36 > async def connect():...
39
40
41 @v1.post("/activations/{activation_name}:trigger")
42 async def trigger_activation(
43     activation_name: str, airflow_client=Depends(AirflowClient)
44 ): ...
48
49
50 @v1.get("/configs", response_model=list[Config])
51 > async def get_configs(session: AsyncSession = Depends(get_session)): ...
63
64
65 @v1.get("/configs:getLatest", response_model=Config)
66 > async def get_latest_config(session: AsyncSession = Depends(get_session)): ...
77
78
79 @v1.get("/configs/{config_id}", response_model=Config)
80 > async def get_config(config_id: int, session: AsyncSession = Depends(get_session)): ...
90
91
92 @v1.post("/configs", response_model=Config)
93 > async def create_config(config: Config, session: AsyncSession = Depends(get_session)): ...
103
104
105 @v1.get("/activations", response_model=list[Activation])
106 > async def get_activations(session: AsyncSession = Depends(get_session)): ...
118
119
120 @v1.get("/schemas", response_model=dict[str, Any])
121 > async def get_schemas(): ...
127
```

Tightlock - Freedom of Choice



Online Available UI
Simple & Fast

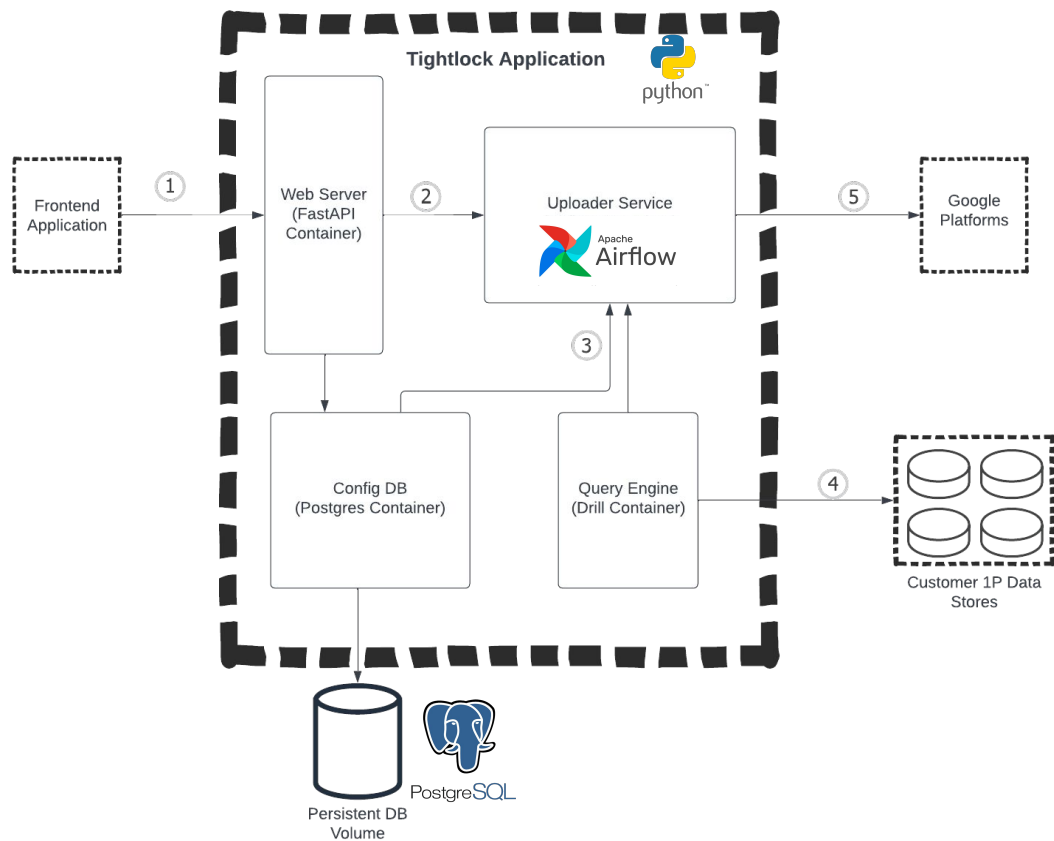


Host Own UI
Flexible

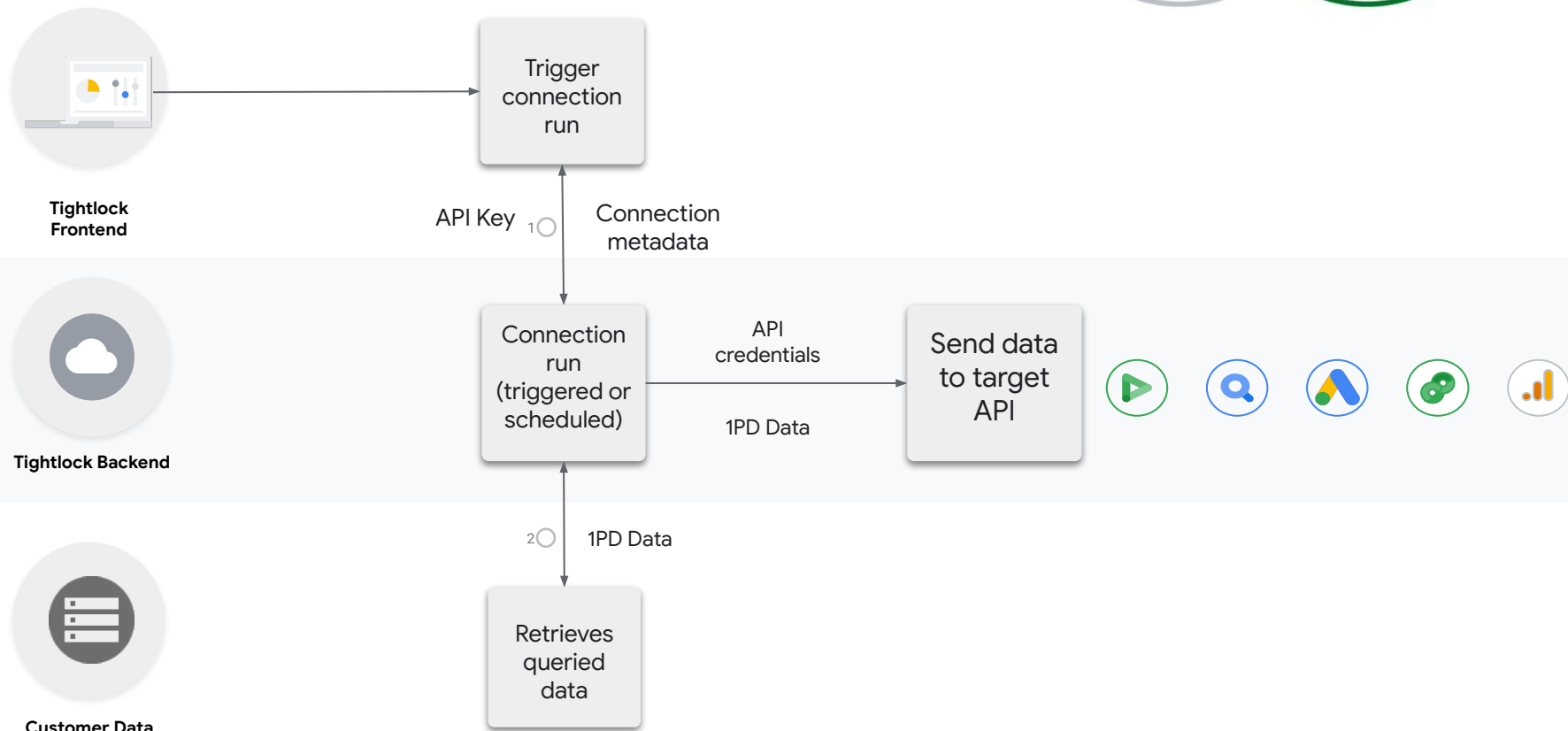


Use API Directly
Advanced

Tightlock - How it Works



Tightlock - How it Works



Once complete, advertisers can run connections and view details

× Connection name Save

i Connection details

Connection name
Salesforce

Data source
Salesforce
[Re-authorize](#)

Schedule
Runs weekly on Mondays at 4:00am UTC
[Edit](#)

🕒 Runs

Most recent x times this connection was run. Includes the next scheduled run and any pending manual runs

Start time	Duration	Status	Run by		Rows imported	Rows with errors
Jan 14, 2021 at 12:38 AM		Scheduled	 Kennen	Scheduled		
Jan 7, 2021 at 12:38 AM	5 minutes	Success	 Zach	Ran manually	1,615	125
Dec 31, 2020 at 12:38 AM	5 minutes	Failed	 Zach	Ran manually	-	-
Dec 24, 2020 at 12:38 AM	5 minutes	Success	 Zach	Ran manually	1,577	110
Dec 17, 2020 at 12:38 AM	5 minutes	Failed	 Kennen	Scheduled	-	-

Rows per page: 10 1-5 of 5 |< < > >|

Clients are then able to see more details on the connection and it's activations, in the connection page.



Extensible interface for sources and destinations

```
@runtime_checkable
class SourceProto(Protocol):

    def __init__(self, config: Dict[str, Any]):
        ...

    def get_data(
        self,
        connections: Sequence[Mapping[str, Any]],
        fields: Sequence[str],
        offset: int,
        limit: int,
    ) -> List[Mapping[str, Any]]:
        ...

    def schema(self) -> Dict[str, Any]:
        ...

    def validate(self) -> ValidationResult:
        ...
```

```
@runtime_checkable
class DestinationProto(Protocol):

    def __init__(self, config: Dict[str, Any]):
        ...

    def send_data(self,
        input_data: List[Mapping[str, Any]],
        dry_run: bool
    ) -> Optional[RunResult]:
        ...

    def fields(self) -> Sequence[str]:
        ...

    def schema(self) -> Dict[str, Any]:
        ...

    def batch_size(self) -> int:
        ...

    def validate(self) -> ValidationResult:
        ...
```



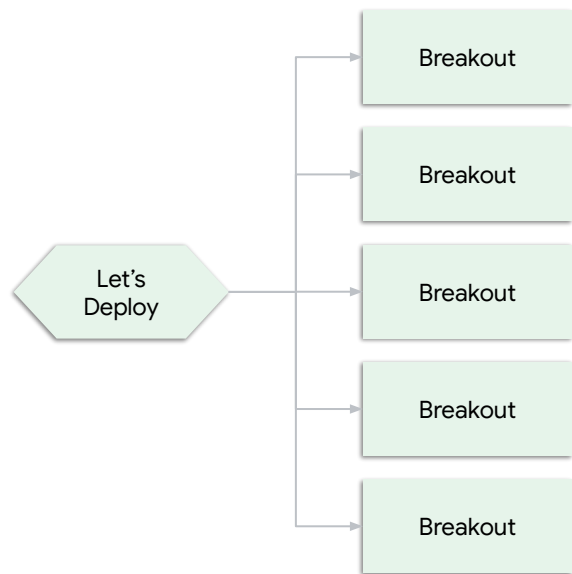
Hack-A-Tech Track

Breakout

Katerina



Breakout Room Set up



Alberto



Inbar



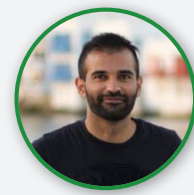
Miguel



Max



Kevin



Ehsaan



Kuba



Jakub



Niels



Tom



Caio



What are we going to do

1. Using Google Cloud Shell
2. Clone Github repository
3. Deploy via Terraform

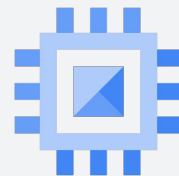


Terraform

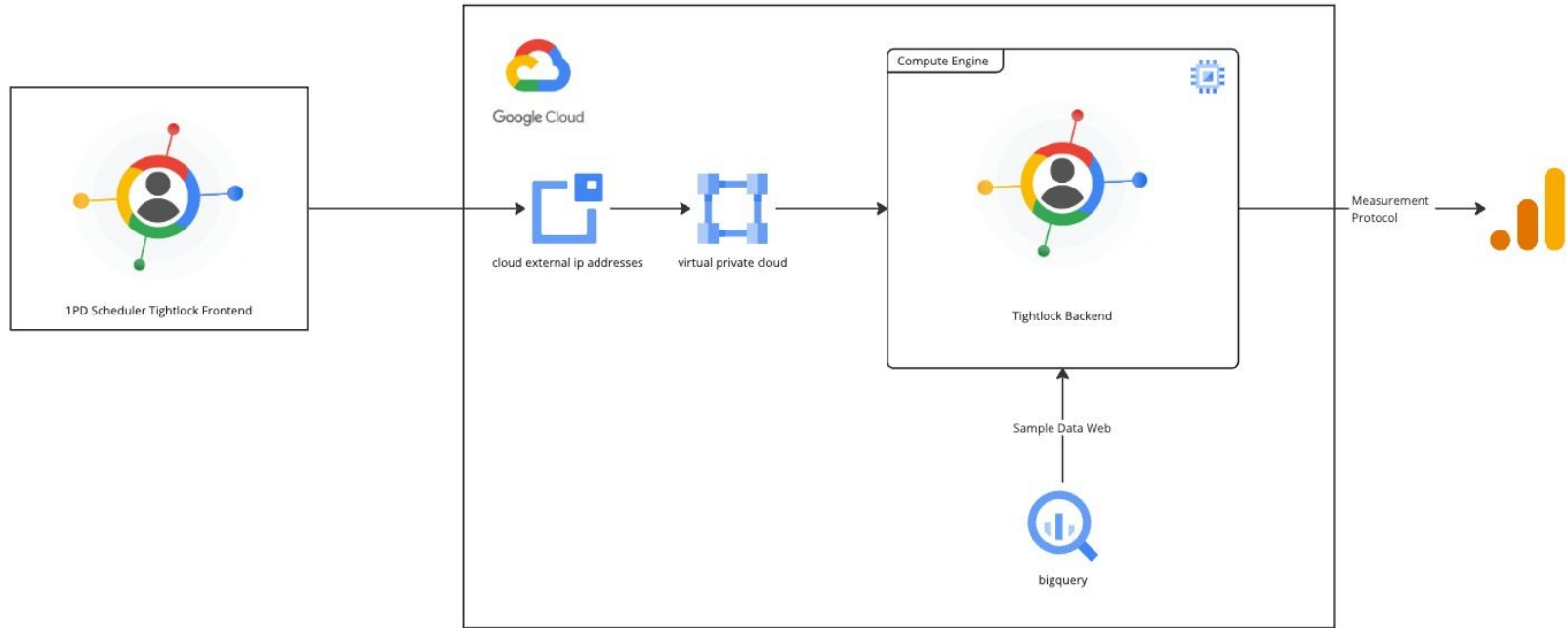


What will be deployed

1. A compute Engine instance
2. A persistent disk connected to the VM
3. A static external IP for the VM
4. BigQuery Tables with sample data



End to End





Hack-A-Tech Track

Deployment in Breakouts

Back to main room at 12:25 CET



Github

We're starting the deployment directly from Github.

Go to github.com/google/Tightlock.

1. Navigate to the `installer` directory.
2. Click on the blue `OPEN IN GOOGLE CLOUD SHELL` button.

This will open Cloud Shell in GCP.

main Tightlock / installer / ↑ Top

Terraform

Preparing for deployment

This installer is using [Terraform](#) to define all resources required to set up Tightlock on Google Cloud Platform. Click the below button to clone this repository in Google Cloud and get started.

 OPEN IN GOOGLE CLOUD SHELL

Update variables.tfvars file

Update the `variables.tfvars` file and provide the following two values. If you clicked the `Open in Cloud Shell` button this file should be opened directly in your Cloud Shell Editor.

variable	description
<code>api_key</code>	API Key for generating the front end connection key.
<code>project_id</code>	Google Cloud Project ID
<code>create_bq_sample_data</code>	If set to true a BigQuery dataset and tables will be created with sample data uploaded via a Cloud Bucket

Deploy

To deploy Tightlock via Terraform run the below two commands in Cloud Shell Terminal.

The `init` command will initialise the Terraform configuration.

```
terraform init
```

The `apply` command will apply the resources to GCP, all required changes are listed when you run this command and you'll be prompted to confirm before anything is applied.

```
terraform apply -var-file=variables.tfvars
```

Destroy deployed resources

In case you want to undo the deployment you can run the below command from the same



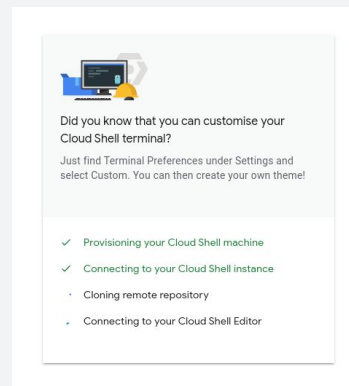
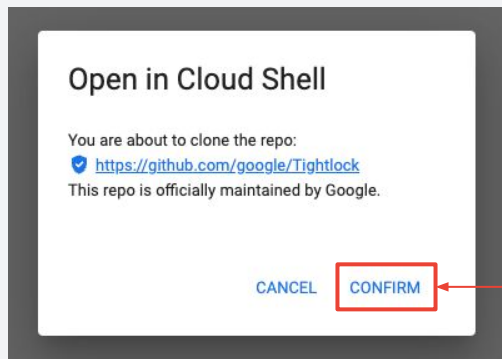
Cloud Shell

Before cloning the repository in your Cloud Shell environment it will ask you to **confirm** this action.

1. Click **CONFIRM**.

A Cloud Shell environment will be provisioned and the code will be cloned in a **cloudshell_open** directory.

It will automatically open the **Cloud Shell Editor** environment.





Terraform: Step 1

For deployment we are using [Terraform](#). An open-source infrastructure-as-code software tool created by HashiCorp.

The previous step opened the correct directory in the Tightlock repository.

We need to fill in a couple of **variables** for this deployment to work. The **variables.tfvars** should be open in your editor.

```
Cloud Shell Editor
File Edit Selection View Go Run Terminal Help

EXPLORER
OPEN EDITORS
NIELSOVERWIJN
cloudshell_open/...
  dags
  installer
    cloud-config.y...
    locals.tf
    main.tf
    README.MD
    sample-data.tf
    variable.tf
    variables.tfvars
  integration_tests
  sample_data
  tightlock_api
    airflow_require...
    airflow.Dockerfile
    CONTRIBUTIN...
    create_env.sh
    docker-compos...
    fastapi_require...
    fastapi.Dockerfile
    LICENSE
  GO

variables.tfvars x
cloudshell_open > Tightlock > installer > variables.tfvars > ...
1 api_key = "value"
2 project_id = "value"
3 create_bq_sample_data = false
4 # compute_engine_zone = "europe-west6-a"
5 # compute_address_region = "europe-west6"
6

cloudshell Output x tight EditorConfig
Homebrew master 0 Cloud Code Connect to Google Cloud Ln 1, Col 1 LF UTF-8 Space
cloudshell x + -
Dj"
nielsoverwijn@cloudshell:~$ cloudshell_open --repo_url "https://github.com/google/Tightlock"
--page "editor" --open_in_editor "installer/variables.tfvars" --git_branch "main" --force_ew_clone
2023/06/16 07:11:16 Cloning https://github.com/google/Tightlock into /home/nielsoverwijn/c
udshell_open/Tightlock
Cloning into '/home/nielsoverwijn/cloudshell_open/Tightlock'...
remote: Enumerating objects: 1583, done.
remote: Counting objects: 100% (764/764), done.
remote: Compressing objects: 100% (308/308), done.
remote: Total 1583 (delta 500), reused 643 (delta 435), pack-reused 819
```



Terraform: Step 2

The deployment needs the following two variables:

api_key	API Key for generating the front end connection key.
project_id	Google Cloud Project ID
create_bq_sample_data	Creates BQ tables with sample data

1. Enter an API key you want to use for this **test deployment**.
2. Set the **GCP Project ID**.
3. Set **create_bq_sample_data** to **true**.

```
Cloud Shell Editor
File Edit Selection View Go Run Terminal Help

EXPLORER
OPEN EDITORS
NIELSOVERWIJN
cloudshell_open/...
days
installer
cloud-config.y...
locals.tf
main.tf
README.MD
sample-data.tf
variable.tf
variables.tfvars
integration_tests
sample_data
tightlock_api
airflow_require...
airflow.Dockerfile
CONTRIBUTIN...
create_env.sh
docker-compos...
fastapi_require...
fastapi.Dockerfile
LICENSE

variables.tfvars
cloudshell_open > Tightlock > installer > variables.tfvars > ...
1 api_key = "value"
2 project_id = "value"
3 create_bq_sample_data = false
4 # compute_engine_zone = "europe-west6-a"
5 # compute_address_region = "europe-west6"
6

cloudshell x Output x tigh EditorConfig
Homebrew master Cloud Code Connect to Google Cloud Ln 1, Col 1 LF UTF-8 Space
cloudshell x +
Dj"
nielsoverwijn@cloudshell:~$ cloudshell_open --repo_url "https://github.com/google/Tightlock"
--page "editor" --open_in_editor "installer/variables.tfvars" --git_branch "main" --force
ew_clone
2023/06/16 07:11:16 Cloning https://github.com/google/Tightlock into /home/nielsoverwijn/c
udshell_open/Tightlock
Cloning into '/home/nielsoverwijn/cloudshell_open/Tightlock'...
remote: Enumerating objects: 1583, done.
remote: Counting objects: 100% (764/764), done.
remote: Compressing objects: 100% (308/308), done.
remote: Total 1583 (delta 500), reused 643 (delta 435), pack-reused 819
```

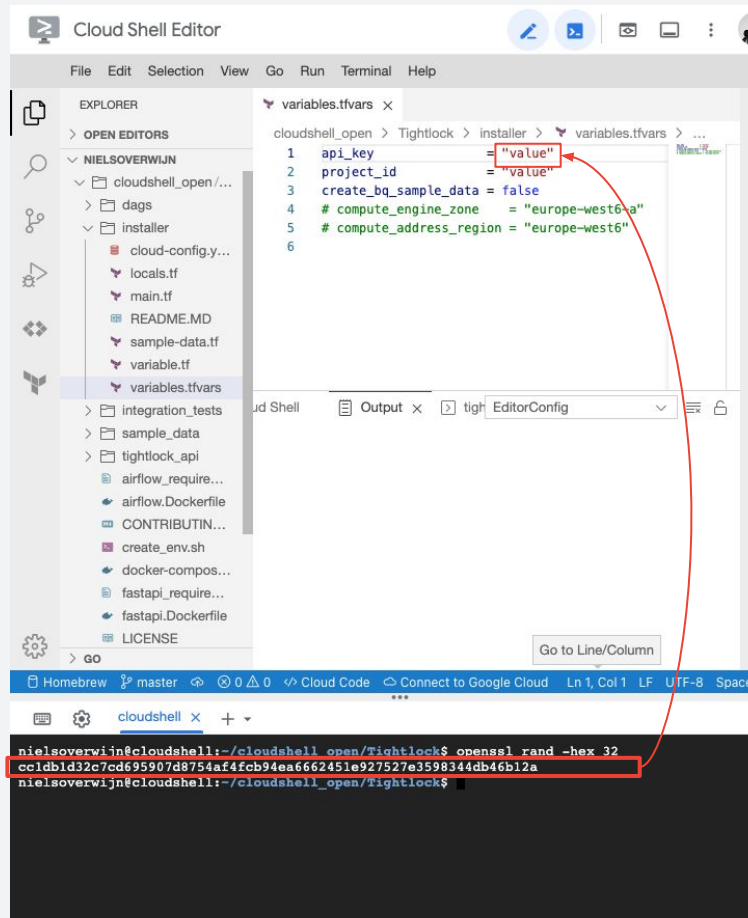


Terraform: Step 3

In case you want to quickly generate a **random** api key you can use the following command:

```
openssl rand -hex 32
```

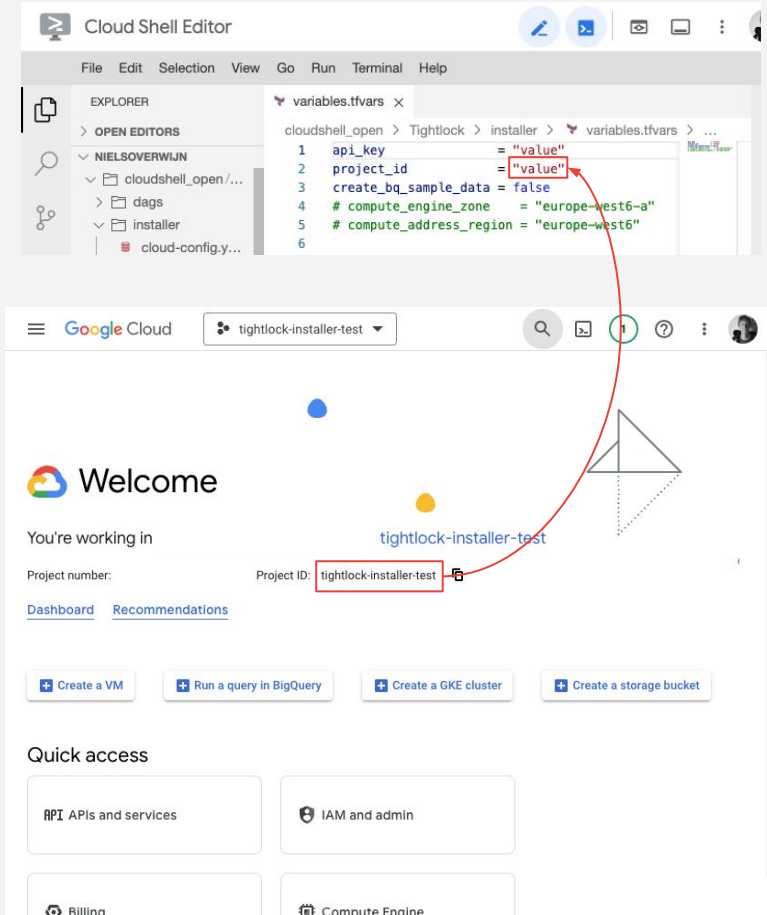
Tip: if you select text in the cloud shell terminal it will automatically be copied to your clipboard.



Terraform: Step 4

In case you don't know your Google Cloud project ID you can navigate to console.cloud.google.com.

The welcome page should list your project ID.

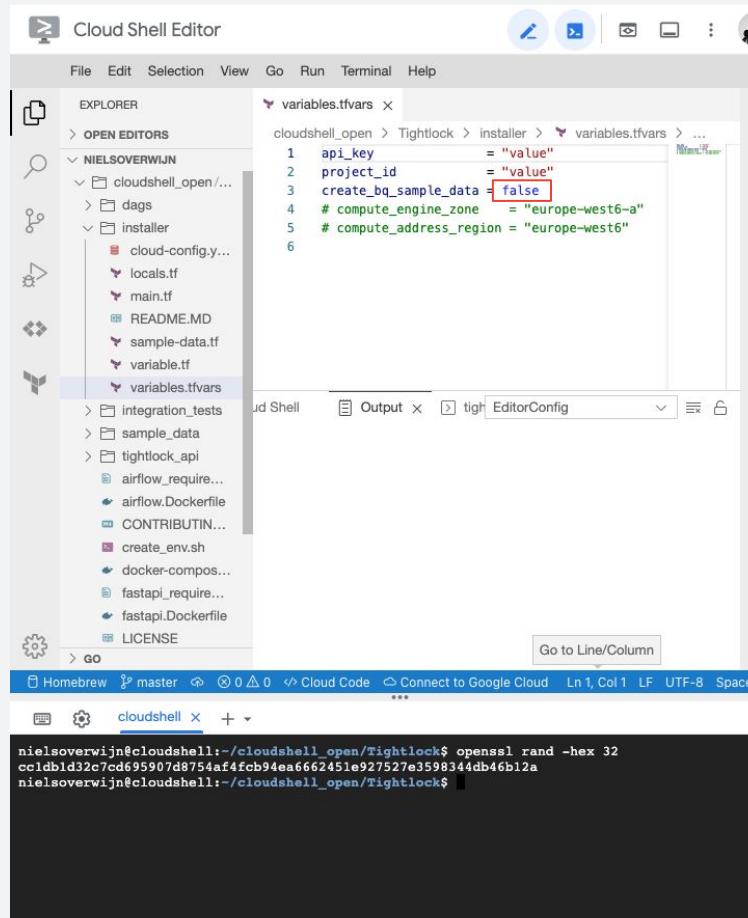




Terraform: Step 5

Set `create_bq_sample_data` to true.

Note: In case you're sharing the same Google Cloud Project only one should set this to `true`.





Terraform: Step 6

Optionally: Uncomment line 4 and 5 if you want the VM to run in Europe.

Default zone is `us-central1-a`. Low CO₂

`eu-west-6` Low CO₂ is a data centre in Zurich, Switzerland, Europe.

For a list of all regions: cloud.google.com/compute/docs/regions-zones

```
Cloud Shell Editor
File Edit Selection View Go Run Terminal Help

EXPLORER
OPEN EDITORS
NIELSOVERWIJN
cloudshell_open/...
  dags
  installer
    cloud-config.y...
    locals.tf
    main.tf
    README.MD
    sample-data.tf
    variable.tf
    variables.tfvars
  integration_tests
  sample_data
  tightlock_api
    airflow_require...
    airflow.Dockerfile
    CONTRIBUTIN...
    create_env.sh
    docker-compos...
    fastapi_require...
    fastapi.Dockerfile
    LICENSE
  go

variables.tfvars x
cloudshell_open > Tightlock > installer > variables.tfvars > com
1  api_key           = "value"
2  project_id        = "value"
3  create_bq_sample_data = false
4  compute_engine_zone = "eu-west-6"
5  compute_address_region = "eu-west-6"
6

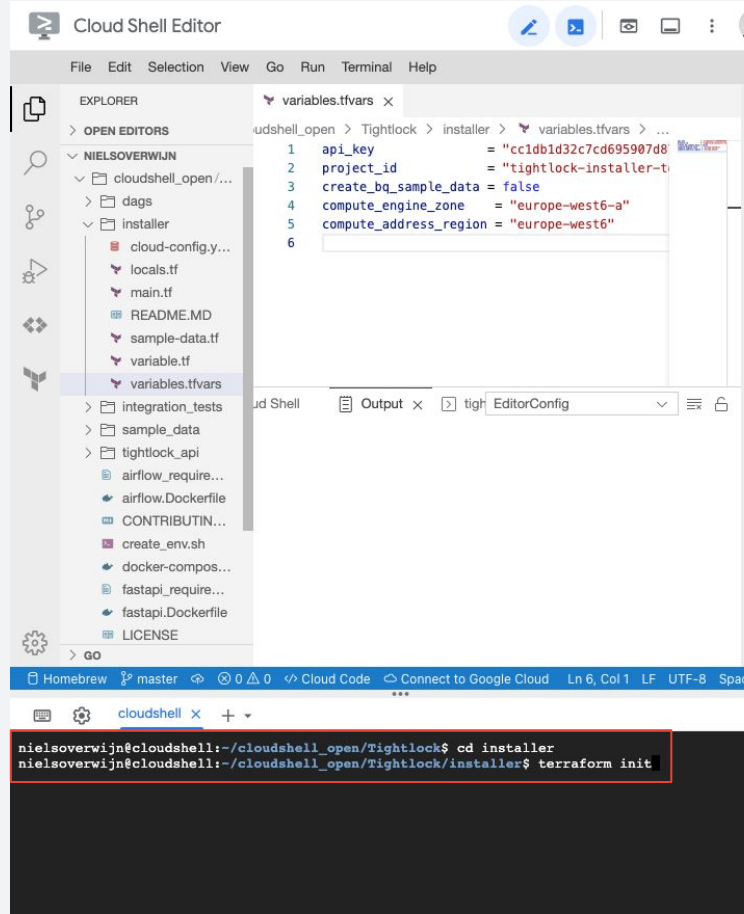
cloudshell | Output x | tigh | EditorConfig
Homebrew master 0 0 0 Cloud Code Connect to Google Cloud Ln 5, Col 1 LF UTF-8
cloudshell x +
nielsoverwijn@cloudshell:~/cloudshell_open/Tightlock$ openssl rand -hex 32
cc1db1d32c7cd695907d8754af4fcb94ea6662451e927527e3598344db46b12a
nielsoverwijn@cloudshell:~/cloudshell_open/Tightlock$
```



Terraform: Step 7

With these variables set we can start the deployment process.

1. Open the **Cloud Shell Terminal**.
2. Run `$ cd installer`
 - This will navigate the terminal to the installer directory.
3. Run `$ terraform init`
 - This will initialize the terraform components needed for this deployment.





Terraform: Step 8

Terraform will let you know when it fully initialized.

```
variables.tfvars
1  api_key      = "cc1db1d32c7cd695907d8"
2  project_id   = "tightlock-installer-t"
3  create_bq_sample_data = false
4  compute_engine_zone = "europe-west6-a"
5  compute_address_region = "europe-west6"
6
```

```
- Installed hashicorp/random v3.5.1 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
nielsovervijn@cloudshell:~/cloudshell_open/Tightlock/installer$
```



Terraform: Step 9

We'll now actually apply all the resource changes in Google Cloud. Before Terraform will apply any changes it will list all resource changes planned and prompt for a confirmation.

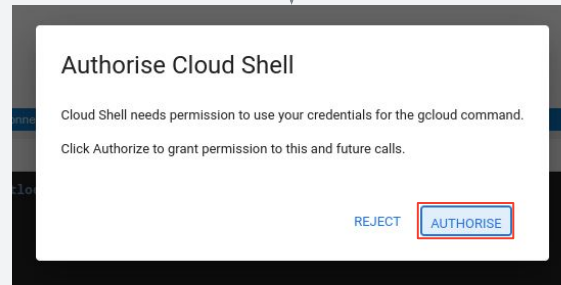
1. Run `$ terraform apply -var-file=variables.tfvars`
 - a. This will start the deployment process
2. Click the **AUTHORISE** button.
3. Type **yes** in the terminal when prompted.

```
Homebrew master 0 0 Cloud Code Connect to Google Cloud Ln 2, Col 40 LF UTF-8 Spaces: 4 terraform
cloudshell x +
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
nielsovervijn@cloudshell:~/cloudshell_open/Tightlock/installs$ terraform apply -var-file=variables.tfvars
```



```
Homebrew master 0 0 Cloud Code Connect to Google Cloud Ln 2, Col 40 LF UTF-8 Spaces: 4 terraform
cloudshell x +
+ service = "compute.googleapis.com"
}

Plan: 4 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ Address = (known after apply)
+ ConnectionCode = (known after apply)

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
Enter a value: yes
```



Terraform: Step 10

Wait ;)

This might take a couple of minutes to complete.

The screenshot shows a VS Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure for 'tightlock' with files like 'variables.tfvars', 'main.tf', and 'README.MD'. The terminal shows the output of a Terraform apply command, indicating the successful creation of various resources like 'google_project_service.cloudresourcemanager', 'google_compute_address.vm-static-ip', and 'google_compute_instance.tightlock-backend'.

```
variables.tfvars
1  api_key      = "cc1db1d32c7cd695907d8"
2  project_id   = "tightlock-installer-t
3  create_bq_sample_data = false
4  compute_engine_zone = "europe-west6-a"
5  compute_address_region = "europe-west6"
6

cloudshell_open > Tightlock > installer > variables.tfvars > ...
1  api_key      = "cc1db1d32c7cd695907d8"
2  project_id   = "tightlock-installer-t
3  create_bq_sample_data = false
4  compute_engine_zone = "europe-west6-a"
5  compute_address_region = "europe-west6"
6

cloudshell > Output x > tight EditorConfig
```

```
google_project_service.cloudresourcemanager: Creating...
google_project_service.compute: Creation complete after 3s [id=tightlock-installer-test/compute.googleapis.com]
google_project_service.cloudresourcemanager: Creation complete after 3s [id=tightlock-installer-test/cloudresourcemanager.googleapis.com]
data.google_compute.default_service_account.default: Reading...
google_compute_address.vm-static-ip: Creating...
google_compute_disk.tightlock-storage: Creating...
data.google_compute.default_service_account.default: Read complete after 0s [id=projects/tightlock-installer-test/serviceAccounts/817494075637-compute@developer.gserviceaccount.com]
google_compute_disk.tightlock-storage: Creation complete after 1s [id=projects/tightlock-installer-test/zones/europe-west6-a/disks/tightlock-6cm4-storage]
google_compute_address.vm-static-ip: Creation complete after 2s [id=projects/tightlock-installer-test/regions/europe-west6/addresses/tightlock-6cm4-static-ip]
google_compute_instance.tightlock-backend: Creating...
google_compute_instance.tightlock-backend: Still creating... [10s elapsed]
```



Terraform: Step 11

Done.

The resources should now be deployed and live. It might take a couple more minutes for the VM to be fully up and running.

Copy and make note of the **API Key**, **Connection Code** and **Address**, you will need this the next steps.

Tip: Running the `terraform output` command will output the Connection Code again.

The screenshot shows a VS Code editor with the following components:

- Explorer:** A file tree on the left showing the project structure. The file `variables.tfvars` is selected.
- Editor:** The `variables.tfvars` file is open, showing the following content:

```
1 api_key      = "cc1db1d32c7cd695907d8"
2 project_id   = "tightlock-installer-t
3 create_bq_sample_data = false
4 compute_engine_zone = "europe-west6-a"
5 compute_address_region = "europe-west6"
6
```
- Terminal:** The terminal at the bottom shows the output of the `terraform apply` command. It indicates that resources were successfully created. The output includes:

```
taller-test/zones/europe-west6-a/disks/tightlock-6cm4-storage]
google_compute_address.vm-static-ip: Creation complete after 2s [id=projects/tightlock-ins
lles-test/regions/europe-west6/addresses/tightlock-6cm4-static-ip]
google_compute_instance.tightlock-backend: Creating...
google_compute_instance.tightlock-backend: Still creating... [10s elapsed]
google_compute_instance.tightlock-backend: Creation complete after 15s [id=projects/tightl
k-installer-test/zones/europe-west6-a/instances/tightlock-backend-6cm4]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:
Address = "34.65.7.98"
Compute Engine Instance = "tightlock-backend-6cm4"
Connection Code = "eyJhcGlLZXkiOiAiY2MxZG1xZWMyY2dJZDYSNTkwN2Q4U0Y0Y0ZmNiOTRlYTY2NjIONTF
TI3NTI3ZTMlOTgzNDRkYjQ2YjE5YjY5IiwiY2MxZG1xZWMyY2dJZDYSNTkwN2Q4U0Y0Y0ZmNiOTRlYTY2NjIONTF
nielsoverwijn@cloudshell:~/cloudshell_open/Tightlock/installer$
```



Test Tightlock API

Requests to the Tightlock deployments are verified by the API Key that you sent in the **X-API-KEY** header. A quick test to see if the deployment is working can be done with the following commands:

```
export TIGHTLOCK_API_KEY=YOUR_API_KEY_HERE
```

```
export TIGHTLOCK_IP=VM_EXTERNAL_IP_ADDRESS_HERE
```

```
curl -i --request POST \  
  --url http://$TIGHTLOCK_IP/api/v1/connect \  
  --header 'X-API-KEY: '$TIGHTLOCK_API_KEY
```

This should output **HTTP/1.1 200 OK**.

The screenshot shows the Cloud Shell Editor interface. The Explorer pane on the left shows the file structure with 'variables.tfvars' selected. The main editor area shows the content of 'variables.tfvars' with line numbers 1 through 6. The terminal at the bottom shows the execution of the curl command and its output.

```
Cloud Shell Editor  
File Edit Selection View Go Run Terminal Help  
EXPLORER  
OPEN EDITORS  
NIELSEVERWIJN  
  terraform.tfstate  
  variable.tf  
  variables.tfvars  
  integration_tests  
  go  
variables.tfvars  
1 7d8754af4fcb94ea6662451e927527e3598344db46b12a  
2  
3  
4  
5  
6  
Homebrew master 0 0 0 Cloud Code Connect to Google Cloud Ln 1, Col 90 LF UTF-8 Spac  
cloudshell  
nielseverwijn@cloudshell:~$ export TIGHTLOCK_API_KEY=cc1db1d32c7cd695907d8754af4fcb94ea6662  
51e927527e3598344db46b12a  
nielseverwijn@cloudshell:~$ export TIGHTLOCK_IP=34.65.7.98  
nielseverwijn@cloudshell:~$ curl -i --request POST \  
  --url http://$TIGHTLOCK_IP/api/v1/connect \  
  --header 'X-API-KEY: '$TIGHTLOCK_API_KEY  
HTTP/1.1 200 OK  
date: Fri, 16 Jun 2023 07:30:18 GMT  
server: uvicorn  
content-length: 0  
nielseverwijn@cloudshell:~$
```



CREATE INSTANCE

IMPORT VM

REFRESH

HELP ASSISTANT

LEARN

Virtual machines

VM instances

Instance templates

Sole-tenant nodes

Machine images

TPUs

Committed-use discounts

Reservations

Migrate to Virtual Machin...

Storage

Disks

Snapshots

Marketplace

Release notes

<|

INSTANCES

OBSERVABILITY

INSTANCE SCHEDULES

VM instances

Filter Enter property name or value

☐	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
☐		tightlock-backend-lx2q	us-central1-a			10.128.0.8 (nic0)	35.184.80.255 (nic0)	SSH ▾ ⋮

Related actions

HIDE

Explore Backup and DR NEW
Back up your VMs and set up disaster recovery

Monitor VMs
View outlier VMs across metrics like CPU and network

Explore VM logs
View, search, analyse and download VM instance logs

Set up firewall rules
Control traffic to and from a VM instance

Patch management
Schedule patch updates and view patch compliance on VM instances

Load balance between VMs [↗](#)
Set up load balancing for your applications as your traffic and users grow

Show debug panel



RESERVE EXTERNAL STATIC IP ADDRESS

RESERVE INTERNAL STATIC IP ADDRESS

REFRESH



HELP ASSISTANT

HIDE INFO PANEL



VPC networks



IP addresses



Bring your own IP



Firewall



Routes



VPC network peering



Shared VPC



Serverless VPC access



Packet mirroring

ALL

INTERNAL IP ADDRESSES

EXTERNAL IP ADDRESSES

IPV4 ADDRESSES

IPV6 ADDRESSES

Filter Enter property name or value



☐	Name	IP address	Access type	Region	Type ↓	Version	In use by
☒	tightlock-lx2q-static-ip	35.184.80.255	External	us-central1	Static	IPv4	VM instance tightlock-backend-lx2q (Zone us-central1-a)
☐	—	10.128.0.8	Internal	us-central1	Ephemeral	IPv4	VM instance tightlock-backend-lx2q (Zone us-central1-a)

tightlock-lx2q-static-ip

Labels help to organise your resources (e.g. cost_centre:sales or env:prod).

+ ADD LABEL

SAVE

DISCARD CHANGES

Tightlock GCP Resources

Analysis

SQL workspace

Data transfers

Scheduled queries

Analytics Hub

Dataform

Partner Centre

Migration

Assessment

SQL translation

Administration

Monitoring

Capacity management

BI Engine

Release notes

Type to search

Viewing workspace resources.

SHOW STARRED ONLY

tightlock-installer-test

External connections

tightlock_sample_data

ga4_app_sample_data

ga4_web_sample_data

☆

⋮

☆

⋮

☆

⋮

ga4_web... ata

ga4_web_sam...

SCHEMA

DETAILS

LINE

Filter Enter property name or value

☐	Field name	Type
☐	client_id	INTEGER
☐	user_id	INTEGER
☐	event_name	STRING
☐	engagement_time_msec	INTEGER
☐	session_id	INTEGER

EDIT SCHEMA

*Untitled

Untitled

RUN

Query completed.

```
1 SELECT * FROM `tightlock-installer-test.tightlock_sample_data.ga4_web_sample_data` LIMIT 1000
```

Press Alt+F1 for accessibility options.

Query results

SAVE RESULTS

EXPLORE DATA

JOB INFORMATION					RESULTS					JSON					EXECUTION DETAILS				
Row	client_id	user_id	event_name	engagem	Row	client_id	user_id	event_name	engagem	Row	client_id	user_id	event_name	engagem	Row	client_id	user_id	event_name	engagem
1	123	456	offline_purchase		1	123	456	offline_purchase		1	123	456	offline_purchase		1	123	456	offline_purchase	
2	124	457	offline_purchase		2	124	457	offline_purchase		2	124	457	offline_purchase		2	124	457	offline_purchase	
3	125	458	offline_purchase		3	125	458	offline_purchase		3	125	458	offline_purchase		3	125	458	offline_purchase	
4	126	459	offline_purchase		4	126	459	offline_purchase		4	126	459	offline_purchase		4	126	459	offline_purchase	

PERSONAL HISTORY

PROJECT HISTORY

REFRESH Show debug panel

Google Partners

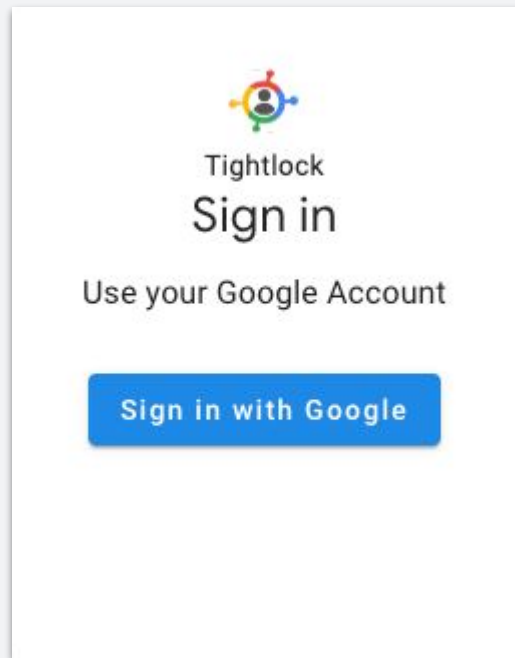


Tightlock UI: Step 1

Navigate to the Tightlock Interface at:

<https://tightlock.dev>

Click on **Sign in with Google**.





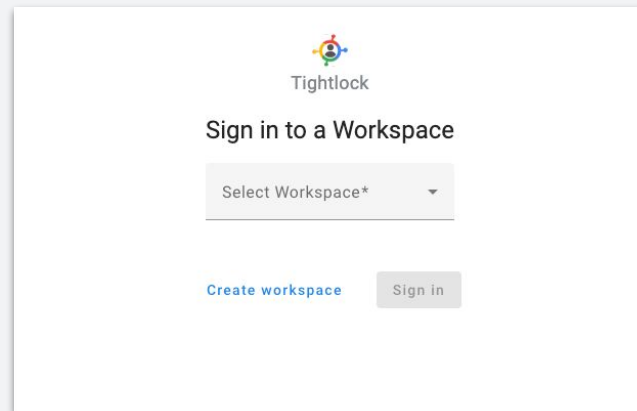
Tightlock UI: Step 2

After login it will ask you to select or create a workspace.

1. Click on **Create workspace**.

Tightlock Workspace: A Tightlock workspace is a single tightlock deployment. This workspace connects to the Tightlock API of your deployment.

Tip: You can give your customer access to the individual workspace created for them, while you have an overview of all your workspaces.





Tightlock UI: Step 3

Provide details of this workspace.

1. Give this workspace a **name**.
2. Paste your **Connection Code** that you've received from the Terraform output before.

Tip: The Connection Code is a Base64 encoded string of:
`{"apiKey": "YOUR API KEY", "address": "IP ADDRESS"}`

Tip: Running the `terraform output` command will output the Connection Code again.

Tightlock

Create a new Workspace

Workspace Name*
Hack-a-Tech Demo

Connection Code*
eyJhcGILZXkiOiAiYTE3N2ExZjIt

Create workspace Back



Tightlock



Hack-a-Tech Demo ▾



Hello Niels Overwijn



Sign out



Connections

Connections



Data Sources



Data Destinations



Run Logs



Settings



Import your own data

Create a new connection to automatically import data from external sources. The connection process will guide you through matching your data with Google data and scheduling future imports. [Learn more](#)

+ CREATE DATA CONNECTION



Tightlock



Hack-a-Tech Demo ▾



Hello Niels Overwijn



Sign out



Connections



Data Sources



Data Destinations



Run Logs



Settings

Data Sources



Import your own data

These are the dialect and location of where your data is coming from.

[+ CREATE DATA SOURCE](#)



Connections



Data Sources



Data Destinations



Run Logs



Settings

Data Destinations



Export your own data

These are the dialect and location of where your data is coming going.

+ CREATE DATA DESTINATION

🔗 Connections

🗄️ Data Sources

📍 Data Destinations

🕒 Run Logs

⚙️ Settings

Run Logs



Nothing running yet

There are no logs captured.

Items per page:

5 ▾

0 of 0





Tightlock



Hack-a-Tech Demo ▾



Hello Niels Overwijn



Sign out



Connections

Manage Members

Manage Workspace



Data Sources



Data Destinations



Run Logs



Settings

Members Settings

+ ADD MEMBER

Add, remove, and edit members associated to your workspaces.

Email

Role



nielsoverwijn@google.com

Owner





Tightlock



Hack-a-Tech Demo ▾



Hello Niels Overwijn



Sign out



Connections

Manage Members

Manage Workspace



Data Sources



Data Destinations



Run Logs



Settings

Workspace Settings

Edit your Workspace connection settings.

Hack-a-Tech Demo

eyJhcGILZXkiOiAiYTE3N2ExZjM1ODExMjY5YTUwNTBIMm(



Analytics: Step 1

Before we send hits to the Measurement Protocol we need to generate a **API Secret** in Google Analytics for this Data Stream.

1. Go to the Google Analytics property you've created for this training ([Pre-work](#)).
2. Open the Data Stream details
3. Click on **Measurement Protocol API secrets**

Web stream details

✓ Data collection is active in the past 48 hours.

Stream details

STREAM NAME	STREAM URL	STREAM ID
demo	https://placeholder.co.uk	5278831441

Events

- Enhanced measurement**
Automatically measure interactions and content on your sites in addition to standard page data from on-page elements such as links and embedded videos may be collected with re ensure that no personally identifiable information will be sent to Google. [Learn more](#)
- Measuring: Page views Scrolls Outbound clicks [+ 4 more](#)
- Modify events**
Modify incoming events and parameters. [Learn more](#)
- Create custom events**
Create new events from existing events. [Learn more](#)
- Measurement Protocol API secrets**
Create an API secret to enable additional events to be sent into this stream through the M

Google tag

- Configure tag settings**
Configure your Google tag behaviour, including cross-domain linking and internal traffic. [Learn more](#)



Analytics: Step 2

1. Click **Review terms** and accept.
2. Click on **Create** in the Measurement Protocol API secrets window
3. Give this new secret a name and click **Create**
4. Copy the **Secret value**, you'll need this in a couple of steps

 Acknowledgement required before creating an API secret [Review terms](#)



Measurement Protocol API secrets

Create a secret to enable additional events to be sent into this stream through the Measurement Protocol. [Learn more about measurement protocol API secrets.](#)

API secrets [Create](#)

Nickname	Secret value	Date created
No API secrets yet, click "Create" to create one.		



× Create new API secret [Create](#)

Nickname

Hack-a-Tech Demo



Measurement Protocol API secrets

Create a secret to enable additional events to be sent into this stream through the Measurement Protocol. [Learn more about measurement protocol API secrets.](#)

API secrets [Create](#)

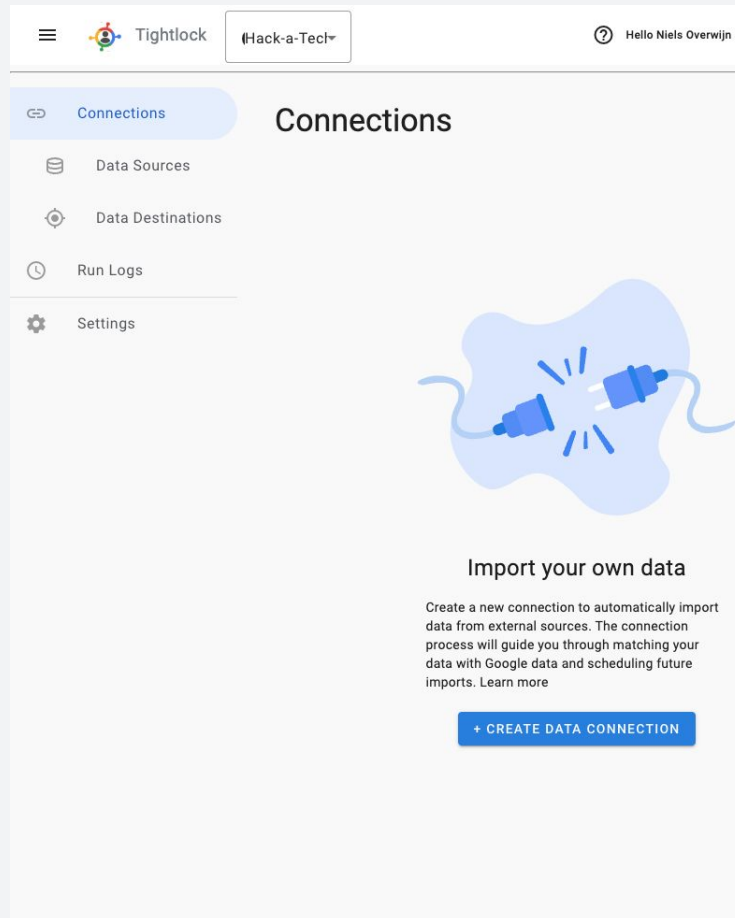
Nickname	Secret value	Date created
Hack-a-Tech Demo	0ZEqlMJwTqiOPkyl2Hy1Tw 	Jun 2, 2023, 3:59:58 PM



Tightlock UI: Step 4

Let's create a new **Connection**.

1. On the **Connections** page click on the **+ CREATE DATA CONNECTION** button.





Tightlock UI: Step 5

We'll connect it to our sample data in BigQuery.

1. Select **BIGQUERY** from the **Data source** drop down.
2. Click **NEXT**.

1PD Scheduler
Tightlock

hack-a-tv

?

Hello Niels Overwijn

Sign out

× Create a Connection

1 Set a Connection Name

ga4_mp_sample

NEXT

2 Select data source

3 Configure data source

4 Select data destination

5 Configure data destination

6 Set a schedule



Tightlock UI: Step 6

We'll connect it to our sample data in BigQuery.

1. Select **BIGQUERY** from the **Data source** drop down.
2. Click **NEXT**.

The screenshot shows the 'Create a Connection' wizard in the Tightlock interface. The top navigation bar includes a menu icon, the text '1PD Scheduler Tightlock', a user profile 'a004', a help icon, a chat icon with 'Hello Niels Overwijn', and a 'Sign out' button. The wizard has a progress indicator on the left with steps 1 through 6. Step 2, 'Select data source', is the current step. It contains a dropdown menu labeled 'Data source' with a downward arrow. Below the dropdown is a panel with two sections: 'COPY FROM EXISTING' and 'NEW'. Under 'NEW', the option 'BIGQUERY' is highlighted. The subsequent steps in the wizard are 'Select data destination', 'Configure data destination', and 'Set a schedule'.

1PD Scheduler
Tightlock

a004

Help icon

Chat icon: Hello Niels Overwijn

Sign out

Create a Connection

1 Set a Connection Name

2 Select data source

Select a data source that is already connected or create a new one

Data source

COPY FROM EXISTING

NEW

BIGQUERY

3 Select data destination

4 Configure data destination

5 Set a schedule



Tightlock UI: Step 7

Provide the following information about the BigQuery Connection:

1. Custom name for this data source
2. Your **Google Cloud Project ID**
3. The BQ Dataset Name: **tightlock_sample_data**
4. The BQ Table Name: **ga4_web_sample_data**
5. Credentials: **Leave blank**
6. **BIGQUERY: BIGQUERY**

Click **VALIDATE** and then **NEXT**.

1PD Scheduler
Tightlock

a004

Hello Niels Overwijn

Sign out

Create a Connection

- Set a Connection Name
- Select data source
- 3 Configure data source**

- 1 Name
- 2 Project ID
- 3 Dataset
- 4 Table
- 5 Credentials
- 6 BIGQUERY



Tightlock UI: Step 8

Now we'll setup the data destination:

1. Select **GA4MP** from the Data destination drop down.
 - a. *This is the GA4 Measurement Protocol destination.*
2. Click **NEXT**.

1PD Scheduler
Tightlock

a004

Hello Niels Overwijn

Sign out

Create a Connection

- Set a Connection Name
- Select data source
- Configure data source
- Select data destination**

Select a data destination that is already connected or create a new one

Data destination

COPY FROM EXISTING

NEW

GA4MP

- Set a schedule



Tightlock UI: Step 9

From the **Select a GA4 Type** select **GA4WEB**.

1PD Scheduler
Tightlock

a004

?

Hello Niels Overwijn

Sign out

×

Create a Connection

✔ Set a Connection Name

✔ Select data source

✔ Configure data source

✔ Select data destination

5 Configure data destination

6 Set a schedule

Select a GA4 Type

GA4APP

GA4WEB



Tightlock UI: Step 10

Fill in the Google Analytics details:

1. A custom name for this data destination
2. API Secret: **YOUR GA4 API Secret Value**
3. The Google Analytics **MEASUREMENT ID**
4. Non Personalized Ads: **Deselected**

Click **VALIDATE** and then **NEXT**.

×

Create a Connection

✓ Set a Connection Name

✓ Select data source

✓ Configure data source

✓ Select data destination

5 Configure data destination

Select a GA4 Type

GA4WEB

1

Name

i

2

Measurement Protocol API Secret

i

3

Measurement ID

i

4

☐ Non Personalized Ads

i

VALIDATE

NEXT



Tightlock UI: Step 11

In the last step we'll set a schedule frequency.

1. Set frequency to **No Schedule**
2. Click **DONE**.

Tip: When selecting **Custom** you can use crontab.guru to find the correct cron expression.

×

Create a Connection

✓

Set a Connection Name

✓

Select data source

✓

Configure data source

✓

Select data destination

✓

Configure data destination

6

Set a schedule

☒ Frequency ☐ Custom

Frequency

No Schedule

HOURLY

WEEKLY

MONTHLY



Tightlock UI: Step 12

The newly created connection should now be listed in the **Connections** page.

1. Click on the **three dot (:)** menu
2. Click on **Run now**.
 - a. This will trigger the activation.

The screenshot shows the Tightlock UI interface. At the top, there's a header with a menu icon, a profile icon, the text '1PD Scheduler Tightlock', a search bar containing 'hack-a-t', a help icon, a chat icon with the text 'Hello Niels Overwijn', and a 'Sign out' button. Below the header, the main section is titled 'Connections' with a '+ ADD DATA CONNECTION' button. Underneath, it says 'These are the connections'. A table lists the connections with columns: Name, Source Type, Destination Type, and Frequency. The table has one entry: 'ga4_mp_sample' with source type 'bq_sample_data' and destination type 'ga4_mp_sample_data'. To the right of the table, a three-dot menu is open, showing options: 'Edit', 'Run now' (highlighted with a red box and a red arrow), and 'Delete'.

Name	Source Type	Destination Type	Frequency
ga4_mp_sample	bq_sample_data	ga4_mp_sample_data	



Connections

Data Sources

Data Destinations

Run Logs

Settings

Run Logs

You can run manually or set a schedule to any you have set, and monitor it here.

Connection Name	Data Source	Destination	Schedule	Status	Run Type	Run At	Rows Imported	Errors
sample_activation	sample_bq_datasource	ga4_mp_web_sample	@monthly	success	manual	2023-06-07T07:32:29.472817+00:00	4	0
sample_activation	sample_bq_datasource	ga4_mp_web_sample	@monthly	success	manual	2023-06-07T07:30:43.354401+00:00	0	0
sample_activation	sample_bq_datasource	ga4_mp_web_sample	@monthly	success	manual	2023-06-07T07:29:44.861416+00:00	0	0

Items per page:

5

1 - 3 of 3

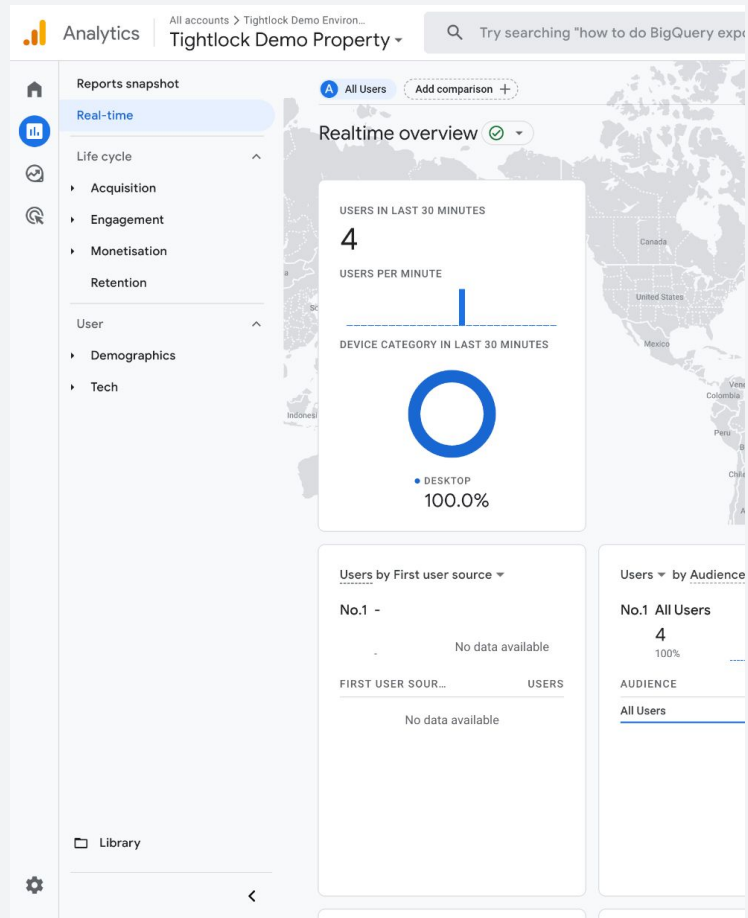




Analytics: Realtime Report

In Google Analytics open the **Real-time** view in the **Reports** tab.

Here, 4 new hits should show up!





Hack-A-Tech Track

Recap & Use Cases

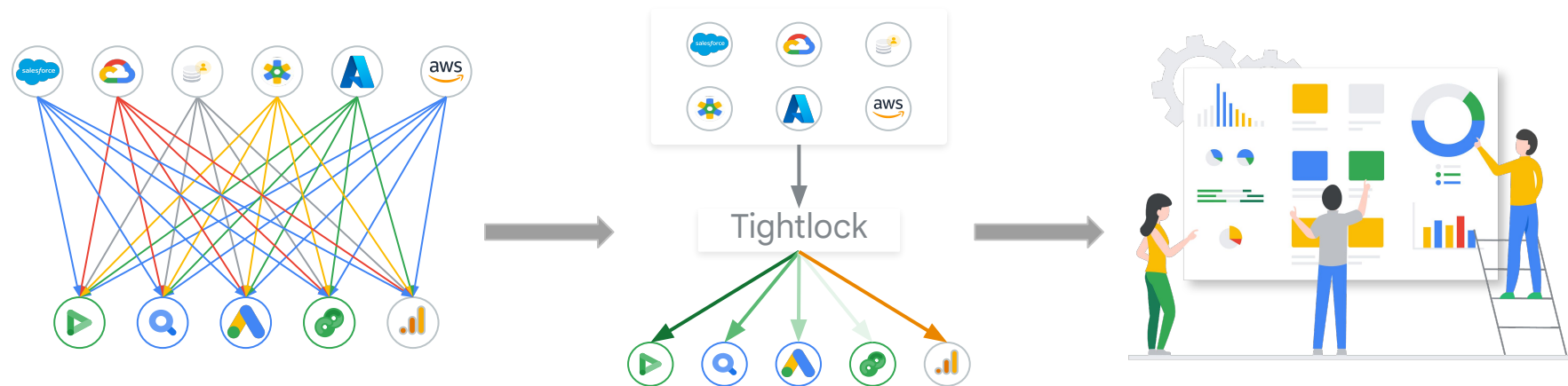
Katerina



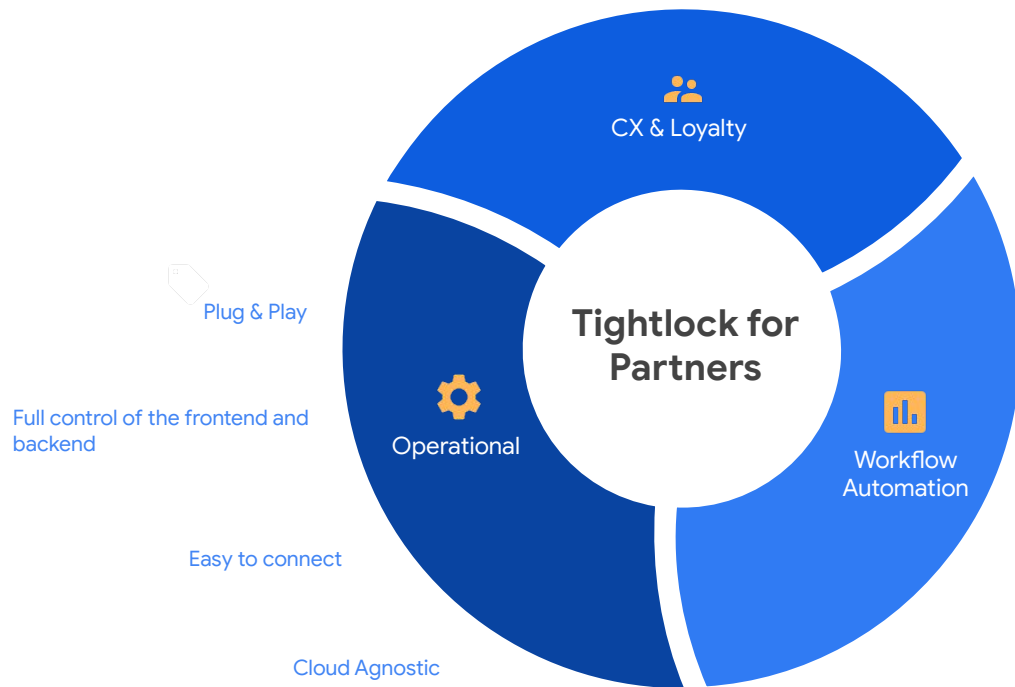
Breakout Recap - Tightlock

- Deployed Tightlock Backend
- Set up a new workspace in Tightlock Frontend
- Created a new connection to the GA4 web stream and execute
- Review the data flowing into GA4

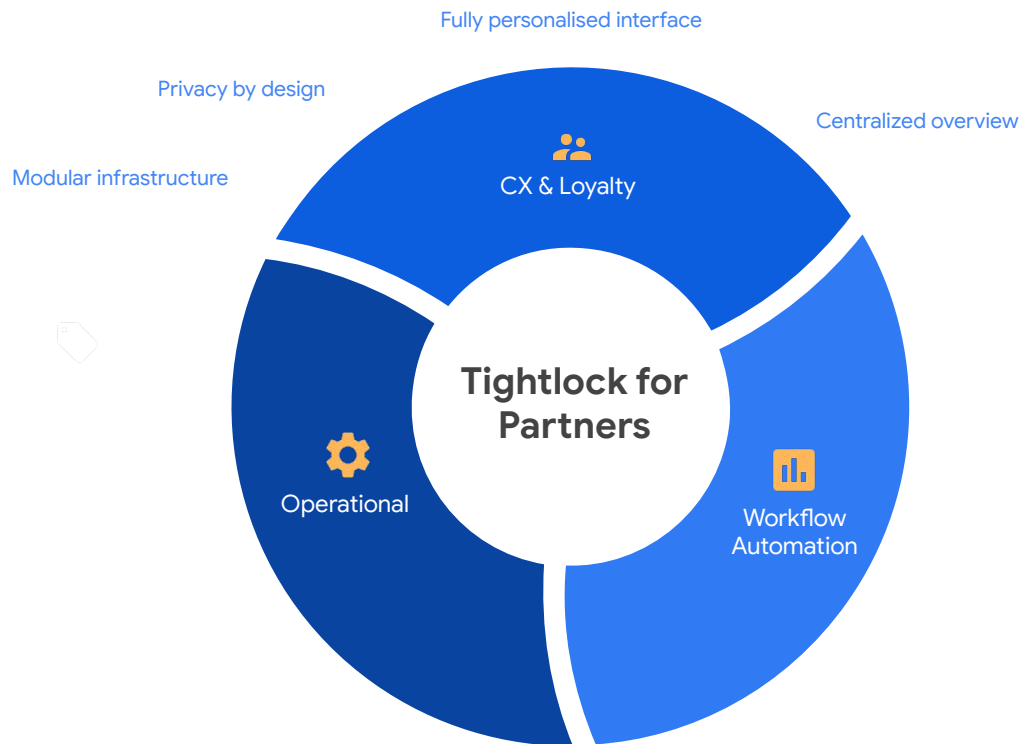
Breakout Recap - Tightlock



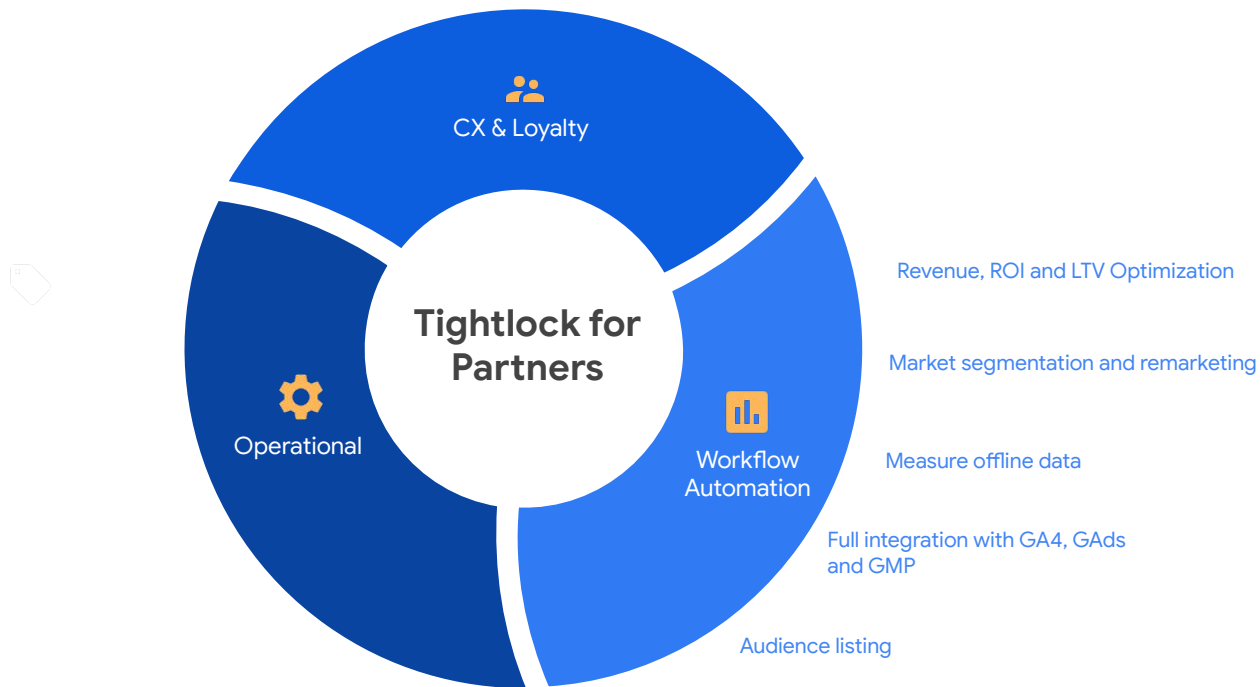
Benefits for Partners



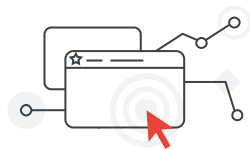
Benefits for Partners



Benefits for Partners



Use cases



Profitability



Measurement



Audiences



Targeting



Hack-A-Tech Track

Roadmap

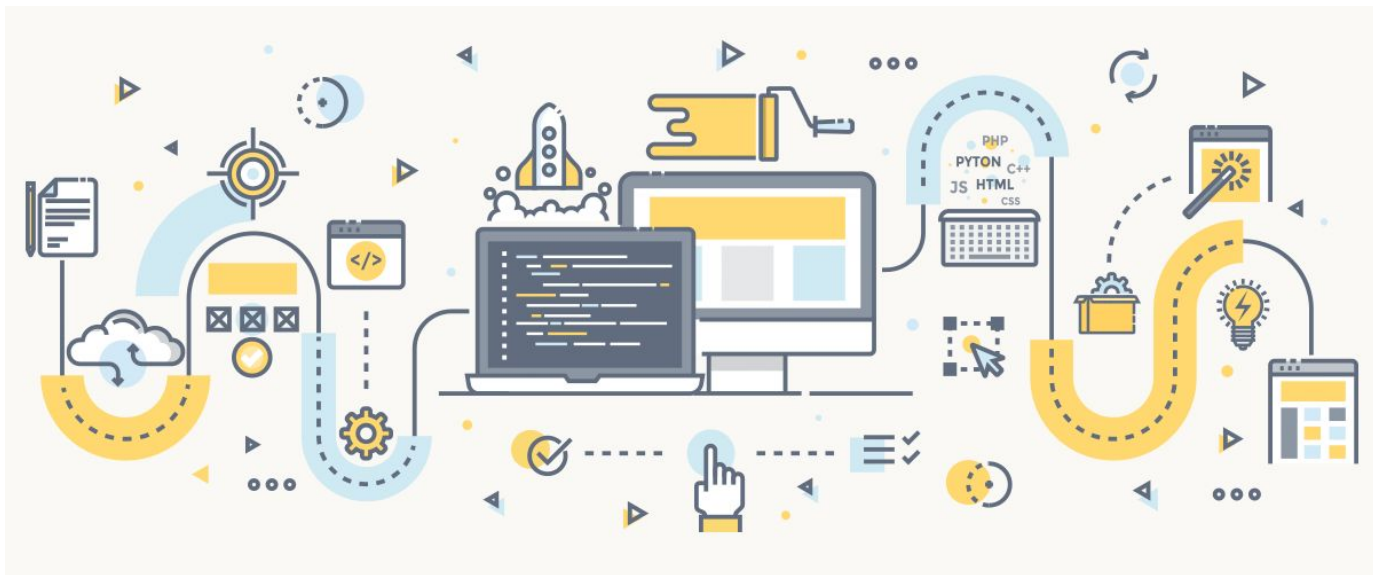
Sarai



Roadmap & Timeline



Automatically Get New Features With Continuous Delivery



Note: this is a high-level view - for more details and previous releases check out [go/tightlock-roadmap](https://cloud.google.com/tightlock/roadmap).



Hack-A-Tech Track

Q&A, Next Steps and Wrap Up

Floris Van de Peppel



Q & A



Feedback & Ideas

Please fill out the **feedback form** and
reach out with further **ideas & questions** to
hack-a-tech@google.com

3 next steps:

1

Bring to Market: Tightlock is a secure and simple 1st party data tool, we brought it early stage to you to further build out and bring innovation to your customers.

2

Start implementing: Use what you learned today and start implementing across your customer portfolio.

3

Talk to your Google team: Keep in touch about the latest updates, implementation and building joint case studies. Reach out to hack-a-tech@google.com or your **partner engagement manager**.

Google Partners



Hack-A-Tech Track

Thank you.



Hack-A-Tech Track

Hack-a-Tech



Hack-A-Tech Track

Hack-a-Tech: Pre-Work

1 Party Data Tool
20th of June

Pre-work Goal

The goal of this pre-work is to get everybody attending the session to the same starting point, and to minimise the chance of encountering any permission issues on the day.

The pre-work is verifying the correct access to Google Cloud Platform and provisioning your Cloud Shell environment.

Please contact hack-a-tech@google.com if you have any questions or issues.



Prerequisites

1. Access to [Google Analytics](#)
2. Access to a [Google Cloud](#) projects with [owner](#) access and [billing](#) enabled.
 - If owner permissions isn't possible, at least the following roles are required:
 - BigQuery Admin
 - Compute Admin
 - Compute Public IP Admin
 - Service Usage Admin





Hack-A-Tech Track

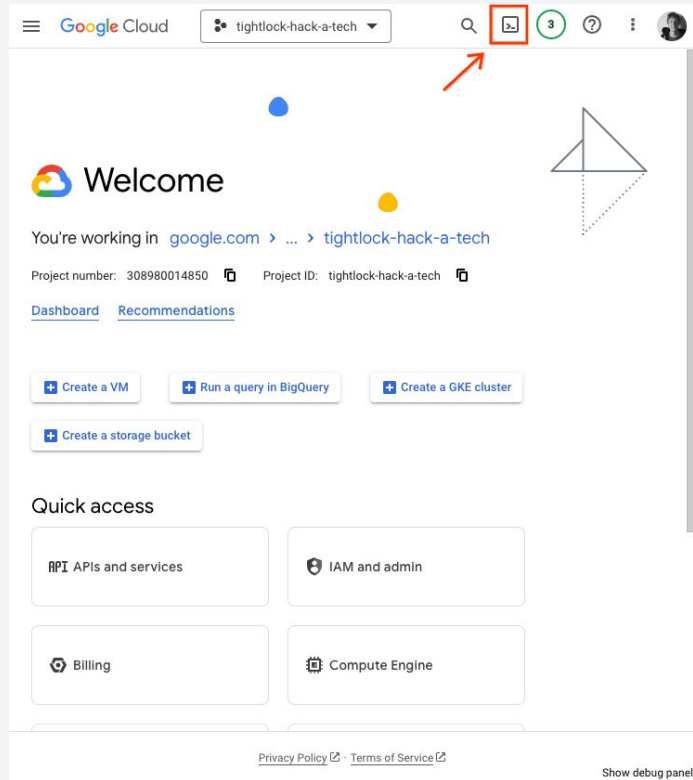
Provisioning Cloud Shell

Prerequisites

- Navigate to console.cloud.google.com.
- Click on the **Terminal Icon** ().

This will provision a **Cloud Shell Machine** for you. The first time you do this it can take up to 20 minutes to fully provision this machine.

More information on [How Cloud Shell Works](#).





Hack-A-Tech Track

Google Analytics



Analytics: Step 1

1. Open [Google Analytics](#)
2. ⚙ Admin -> Create an account

The screenshot displays the Google Analytics Admin interface. At the top, the 'Analytics' logo is followed by the text 'All accounts > My Google Account' and 'A Property'. A search bar is visible on the right. The left sidebar features a vertical menu with icons for Home, Reports, Exploration, Admin, and a gear icon for settings. The main content area is titled 'ADMIN' and shows a 'Create Account' button with a red arrow pointing to it. Below this are links for 'Account Settings', 'Account Access Management', 'All Filters', 'Account Change History', and 'Rubbish Bin'. A red arrow also points to the gear icon in the bottom left of the sidebar.



Analytics: Step 2

1. Give the account a name and press next

The screenshot shows the Google Analytics 'Account setup' interface. At the top, the 'Analytics' logo is on the left, and the 'ADMIN' and 'USER' tabs are on the right, with 'ADMIN' selected. A vertical sidebar on the left contains icons for home, reports, audience, engagement, conversions, and account management. The main content area is titled '1 Account setup' with a blue circle containing the number '1'. Below this, there is a section for 'Account details'. Under 'Account name (Required)', there is a text input field containing 'Demo Account'. A note below the input field states: 'Accounts can contain more than one tracking ID.' A back arrow icon is visible on the left side of the main content area.



Analytics: Step 3

1. Give the property a name and press next
2. Complete the rest of the form and press create

We will be creating two properties, so include “web” in the description to avoid confusion with the “server” property we will be creating soon

Analytics

ADMIN USER

2 Property setup

Property details
A property represents a business's web and/or app data. An account can contain one or more properties. [Learn more](#)
[*Navigate to your Firebase account](#) if you want to create a new property for an existing Firebase project.

Create a Google Analytics 4 property to measure your web and/or app data.

Property name
Demo E-commerce Site (web)

Reporting time zone
United Kingdom (GMT+00:00) GMT

Currency
British Pound (£)

You can edit these property details later in Admin

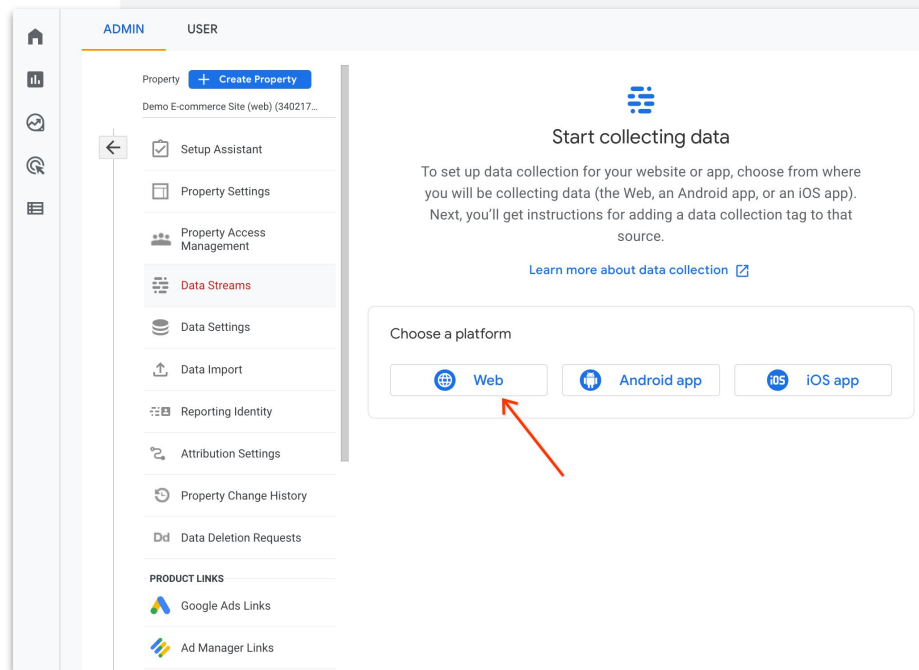
[Show advanced options](#)

Next Previous



Analytics: Step 4

1. Create a web data stream





Analytics: Step 5

1. Add a placeholder URL for now (we will update this shortly after deploying the website), give it a name
2. Press create stream

Set up your web stream

Website URL

http... placeholder.co.uk

Stream name

Demo e-commerce store

 **Enhanced measurement**

Automatically measure interactions and content on your sites in addition to standard page view measurement. Data from on-page elements such as links and embedded videos may be collected with relevant events. You must ensure that no personally identifiable information will be sent to Google. [Learn more](#)

 Page views

 Scrolls

 Outbound clicks

Measuring: + 3 more



Create stream



Analytics: Step 6

1. Make a note of the measurement ID

×

Web stream details

⋮

⚠

Data collection isn't active for your website. If you installed tags more than 48 hours ago, make sure that they are set up correctly.

View tag instructions

Stream details

✎

STREAM NAME	STREAM URL	STREAM ID
Demo e-commerce store	https://placeholder.co.uk	0000000000
MEASUREMENT ID		
G-ABCDEFGHIJ 		

←

Events

✦✦

Enhanced measurement

Automatically measure interactions and content on your sites in addition to standard page view measurement.

Data from on-page elements such as links and embedded videos may be collected with relevant events. You must ensure that no personally identifiable information will be sent to Google. [Learn more](#)

☒

Measuring

Page views

Scrolls

Outbound clicks

2 more

⚙



Hack-A-Tech Track

Thank You