

VERSION 2

DEC 22, 2023

OPEN ACCESS



DOI:

[dx.doi.org/10.17504/protocols.io.q26g7yb3kgwz/v2](https://dx.doi.org/10.17504/protocols.io.q26g7yb3kgwz/v2)

**Protocol Citation:** Ana Mariya Anhel, Lorea Alejaldre, Ángel Goñi-Moreno 2023. OT-2 Media dispensing and culture inoculation protocol. **protocols.io** <https://dx.doi.org/10.17504/protocols.io.q26g7yb3kgwz/v2> Version created by Lorea Alejaldre

**License:** This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

**Protocol status:** Working  
We use this protocol and it's working

## 🌐 OT-2 Media dispensing and culture inoculation protocol V.2

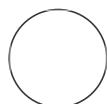
Ana Mariya

Anhel<sup>1</sup>,

Lorea Alejaldre<sup>1</sup>, Ángel Goñi-Moreno<sup>1</sup>

<sup>1</sup>Centro de Biotecnología y Genómica de Plantas, Universidad Politécnica de Madrid (UPM)-Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria (INIA/CSIC), Madrid, Spain

Ángel Goñi-Moreno: [angel.goni@upm.es](mailto:angel.goni@upm.es)



biocomp.cbpp

### ABSTRACT

This protocol is meant to distribute reactive(s) into well plates with a single channel pipette, and then transfer culture samples from source plate(s) to these plates with a multi-channel pipette.

This protocol uses a python script for an Opentrons 2 robot and an excel file containing the required variables to set the number of samples, volumes of transfer, type of plates, etc...

In our laboratory, this protocol has been used as part of the "High-throughput workflow for the genotypic characterization of transposon library variants" also available in protocols.io to prepare cultures plates to perform a subsequent counter-selection

This protocol is a set of instructions or description of the [LAP repository](#) entry **LAP-CellMediaInoculation-OT2-1.0.0**

**Created:** Oct 03, 2023

**Last Modified:** Dec 22, 2023

**PROTOCOL integer ID:** 88734

**Keywords:** automation, opentrons, OT-2, media dispensing, culture inoculation

**Funders**  
**Acknowledgement:**  
Comunidad de Madrid  
Grant ID: Y2020/TCS-6555, 2019-T1/BIO-14053  
MCIN/AEI  
Grant ID: CEX2020-000999-S, PID2020-117205GA-I00  
European Research Council  
Grant ID: 101044360

GUIDELINES

This protocol was run in a python 3.7.1, OT App Software Version 6.3.1 and opentrons API level version 2.14 in a Linux 4.14.74 system (these are the OT-2 specifications). In the script several packages are used: pandas (0.25.3), openpyxl (3.1.2), math, random

It has been tested with cultures from *Pseudomonas putida* KT2440 as part of the High-throughput workflow for the genotypic characterization of transposon library variants. LB and M9-citrate media with either kanamycin, gentamicin, streptomycin or ampicillin haven been successfully run.

The maximum number of 96-well final plates per run given **1 source plate**, 2 different types of tip racks (with the replacement of the tip rack set as True) is **7 final plates** (use 1 falcon tube rack and 2 tip racks)

MATERIALS

Software

- Python 3.7.1
- opentrons software version 6.3.1
- python packages: pandas (0.25.3), openpyxl (3.1.2), math, random
- OT App
- Excel


OT-2 Labware

- Opentrons Tip racks


Equipment	
Opentrons 96 Tip Rack 300 µL	NAME
Tip rack	TYPE
Opentrons	BRAND
-	SKU
<a href="https://labware.opentrons.com/opentrons_96_tiprack_300ul?category=tipRack">https://labware.opentrons.com/opentrons_96_tiprack_300ul?category=tipRack</a>	LINK

Equipment	
<b>Opentrons 96 Filter Tip Rack 20 µL</b>	NAME
Tip rack	TYPE
Opentrons	BRAND
-	SKU
<a href="https://labware.opentrons.com/opentrons_96_filtertiprack_20ul?category=tipRack">https://labware.opentrons.com/opentrons_96_filtertiprack_20ul?category=tipRack</a>	LINK

- 96-well plates

Equipment	
<b>96-well plates, flat bottom, non treated</b>	NAME
Cell culture plates	TYPE
VWR	BRAND
734-2781	SKU
<a href="https://es.vwr.com/store/catalog/product.jsp?catalog_number=734-2781">https://es.vwr.com/store/catalog/product.jsp?catalog_number=734-2781</a>	LINK
	

- Opentrons Falcon Tube Rack

Equipment	
<b>Opentrons 15 Tube Rack with Falcon 15 mL Conical</b>	NAME
OT Tube Rack	TYPE
Opentrons	BRAND
-	SKU
<a href="https://labware.opentrons.com/opentrons_15_tuberack_falcon_15ml_conical/">https://labware.opentrons.com/opentrons_15_tuberack_falcon_15ml_conical/</a>	LINK
	

■ 15mL Falcon tubes

Equipment	
<b>Falcon® Conical Centrifuge Tubes 15mL</b>	NAME
Flaocn Tube	TYPE
Falcon	BRAND
352096	SKU
<a href="https://ecatalog.corning.com/life-sciences/b2c/US/en/Liquid-Handling/Tubes,-Liquid-Handling/Centrifuge-Tubes/Falcon%C2%AE-Conical-Centrifuge-Tubes/p/falconConicalTubes">https://ecatalog.corning.com/life-sciences/b2c/US/en/Liquid-Handling/Tubes,-Liquid-Handling/Centrifuge-Tubes/Falcon%C2%AE-Conical-Centrifuge-Tubes/p/falconConicalTubes</a>	LINK

**Equipment:**

Equipment	
<b>OT-2</b>	NAME
Liquid handler	TYPE
Opentrons	BRAND
OT-2	SKU

Equipment	
<b>HEPA Module</b>	NAME
Opentrons	BRAND
OT-2-HEPA	SKU
<a href="https://opentrons.com/modules/hepa-module/">https://opentrons.com/modules/hepa-module/</a>	LINK

Equipment	
<b>Single Channel Electronic Pipette (GEN2) 300uL</b>	NAME
Opentrons Pipette	TYPE
Opentrons	BRAND
-	SKU
<a href="https://shop.opentrons.com/single-channel-electronic-pipette-p20/">https://shop.opentrons.com/single-channel-electronic-pipette-p20/</a>	LINK

Equipment	
8 Channel Electronic Pipette (GEN2) 20uL	NAME
Multi channel pipette	TYPE
Opentrons	BRAND
-	SKU
<a href="https://shop.opentrons.com/8-channel-electronic-pipette/">https://shop.opentrons.com/8-channel-electronic-pipette/</a>	LINK

## SAFETY WARNINGS



It is important to use HEPA module to work in sterility

## BEFORE START INSTRUCTIONS

Note that the source and final 96-well plates will follow the same order (If the wells of a column of the source plates has empty wells the multichannel will aspirate in that well) but media will not be dispensed in those corresponding final wells

# Files Preparation

## 1 Preparing Customized Template

Preparing the template (a .xlsx) with the specific variables for each experiment and a .pdf that contains the instructions on how to fill the template

Here we attach one Excel with several sheets:

1. **GeneralVariables:** variables related mainly to the labware that is going to be used
2. **PipetteVariables:** variables related to the pipettes that are going to be used
3. **PerPlateVariables:** variables associated with the specifications of each source plate

 Template-VariablesPlateIncubation.xlsx 17KB

 InoculationInstructions.pdf 179KB

### 1.1 *Fill the template with the corresponding values*

## 1.2 Store it as *VariablesPlateIncubation.xlsx*

### Note

The file should be spelt **precisely** as *VariablesPlateIncubation.xlsx* or the Python script won't read it

## 2 Transferring file to Robot

Transfer the *VariablesPlateIncubation.xlsx* to the directory */data/user\_storage* of the OT system that we will use to perform the protocol.

### Note

Before transferring any file to the OT, we need to know the IP of the robot.

This can be obtained in the Networking section of the Device that we are going to

**OT-App -> Devices -> Chosen Robot (three dots) --> Robot Settings --> Networking**

In this tab, you can see 2 types of IP; one is shown if both the robot and you are connected to the same Wifi, and the other is shown if the computer and the robot are connected via USB. Both connections can be used for this step.

### Note

An OT-key should have been previously generated to connect to the robot, and it is done with the *ssh-keygen* command and **transferring the public key to the OT**.

For more information about how to generate and set the connection between your computer and the Opentrons robot, visit <https://support.opentrons.com/s/article/Setting-up-SSH-access-to-your-OT-2>

Here, we present a summary of how to transfer the files in 3 OS: *Windows*, *MacOS* and *Linux*.

### MacOS/Linux

We will use the command line with scp to transfer the file *VariablesPlateIncubation.xlsx* to the OT system.

We need to perform the following command

#### Command

##### File passing from linux (our computer) to linux (OT raspberry)

```
scp -i [ot_key path] [file path] root@[IP_OT]:/data/user_storage
```

#### Note

You could face difficulties transferring files in the MacOS Ventura (13) and Sonoma (14). These problems can be solved by adding the argument -O (uppercase o) to the command

#### Command

##### Transferring files to OT (MacOS 13 and 14)

```
scp -Oi [ot_keypath] [file path] root@[OP_robot]:/data/user_storage
```

### Windows

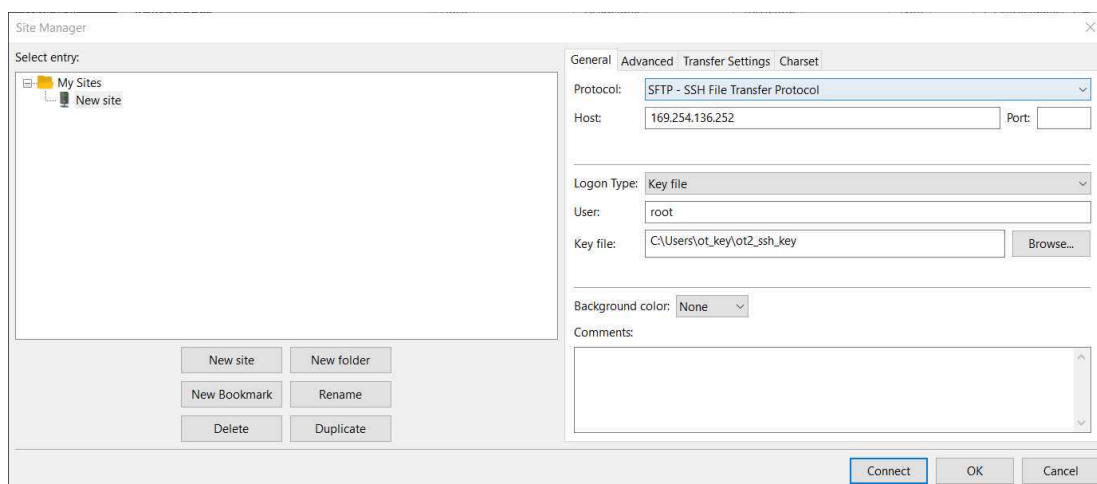
There are several ways to send files from a Windows to a Linux (for example, with a virtual machine or Windows Powershell in the latest versions of Windows).

Here, we will use **FileZilla** (<https://filezilla-project.org/download.php?type=client>).

Go to **File -> Site Manager -> New Site -> Change Protocol to SFTP**. Then, introduce in Host the OT IP, change *Logon Type* to key file, change *User* to root and give the directory where the OT-key file is. The



final window should look something like the following picture



Example of setting the FileZilla to transfer files from Windows (our computer) to Linux (OT)

Then press *Connect*, and we will have a connection between our computer and the robot.

After this connection, we should be able to transfer *VariablesPlateIncubation.xlsx* (in our computer) to the directory */data/user\_storage* in the robot.

This method of transferring files can also be done in Linux and MacOS.

#### Note

Take into account that the IP of the robot could change, so it is possible that the host in these connections will need to be changed from time to time.

### 3 Adding the custom labware



There is only a need to do this step when the labware you are using is not OT official or included in the OT app.

#### 3.1 Creation of .json file

The description file can be obtained by describing the labware dimensions at <https://labware.opentrons.com/create/>

### 3.2 *Uploading files to the OT-app*

In the OT app, we need to perform the following route: **Labware -> Import -> Choose File -> Select file we have created in step 3.1**

#### Expected result

After uploading the labware, you should be able to see the new labware in the Labware tab of the OT App. All custom labware can be found more quickly in the category *Custom Labware*.

### 3.3 *Transfer labware files to the robot*



If you are using the entry **LAP-CellMediaInoculation-OT2-1.0.0** and **custom labware**, an additional step is needed, which is transferring a folder with the custom labware.

We need to create for our custom labware a folder with the API name containing the description file (.json) called 1.json and then transfer that folder to the robot's folder `/data/labware/v2/custom_definitions/custom_beta` in a similar way as in the Step 2 but with the difference that is a directory that needs to be transferred and not a file.

#### Command

##### Transferring the custom labware to OT (Linux)

```
scp -i [ot_key] -r [directory_custom_labware] root@[IP_OT]:/data/labware/v2/custom_definitions/custom_beta
```

#### Note

We do not need to execute this part every time the protocol is used, only when that labware is not included in the OT official labware and these directories are not in the robot.

## Prepare Robot OS

### 4 Install needed packages




This script needs the package *openpyxl*, which is not installed by default in the OT-2 robots

#### Note

This step is only needed if the package is not installed in the robot.

If the package is not installed, an error will appear when running the script in the robot. This error will appear while simulating the script in the app, but you can ignore it.

## 4.1 *Connect to the robot*

 **go to step #2** to find the IP of the robot in which you want to run the script

To connect to the robot, you can do it via ssh with the following command

#### Command

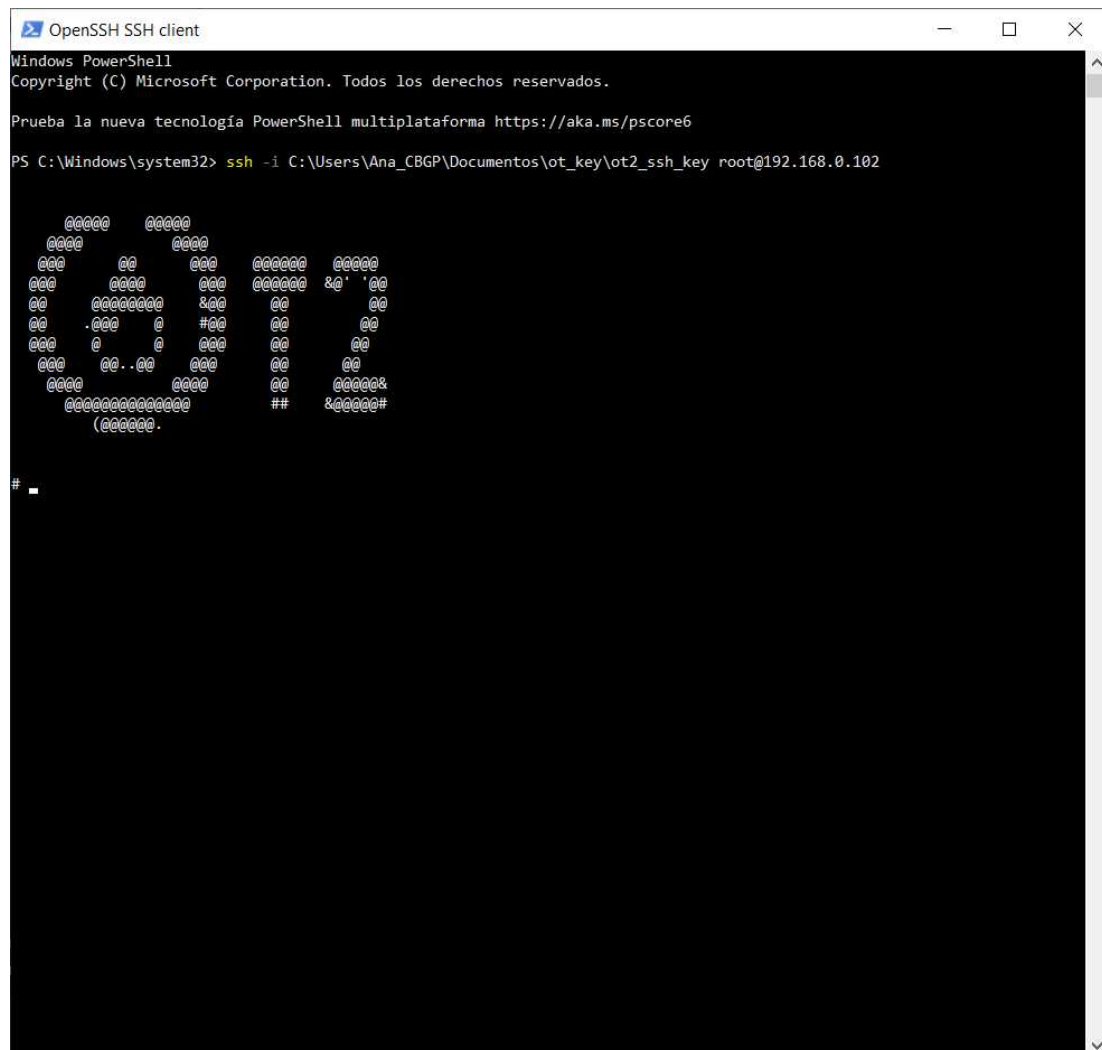
##### Connect to Linux based OT via ssh

```
ssh -i [path ot_key] root@[Robot_IP]
```

In Windows, you can do this command in Windows Powershell

## Expected result

If the connection has been successful, you should obtain a screen similar to the following image



```
OpenSSH SSH client
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Windows\system32> ssh -i C:\Users\Ana_CBP\Documentos\ot_key\ot2_ssh_key root@192.168.0.102

      @@@@@ @@@@@
     @@@@  @@@@
    @@@  @  @@@ @@@@ @@@@
   @@@ @@@@ @@@ @@@@ @@@
  @@@ @@@@ @@@ @@@@ @@@
 @@@ @@@@ @@@ @@@@ @@@
@@@ @@@@ @@@ @@@@ @@@
 @@@ @@@@ @@@ @@@@ @@@
  @@@ @@@@ @@@ @@@@ @@@
   @@@ @@@@ @@@ @@@@ @@@
    @@@ @@@@ @@@ @@@@ @@@
     @@@ @@@@ @@@ @@@@ @@@
      @@@@ @@@@
    (@@@@@.

#
```

Drawing obtained when entering an OT-2 system

## 4.2 *Install the package*

Once inside the robot's system, you need to run the following command

#### Command

#### Install openpyxl package (Linux 4.14.74-v7)

```
pip install openpyxl
```

#### Note

For more information about installing packages in the opentrons robots, check the following Opentrons page: <https://support.opentrons.com/s/article/Using-Python-packages-in-Python-API-protocols>

## Run Protocol

### 5 Load script in OT-App

Now that we have transferred the variable files to the robot, we can load the script and run it in the selected robot

#### Note

This whole step has been developed with version 6.3.1 of the OT-App and has been tested in versions up to 7.0.2

**Indications may vary from version to version of the opentrons App and the version of the script.**

## Software

### Opentrons App

NAME

Windows >=10, Mac >=10 , Ubuntu >=12.04

OS

Opentrons

DEVELOPER

<https://opentrons.com/ot-app/>

SOURCE  
LINK

## 5.1 *Load the script in the App*

### Protocols -> Import -> Drag Python script

#### Note

The last script version can be found at

<https://github.com/BiocomputationLab/LAPrepository/tree/main/LAPEntries> (the name of this file is the user's choice) in the directories with the name **LAP-CellMediaInoculation-OT2** followed by the version.

As well we can find the latest version of the script at

<https://www.laprepo.cbgp.upm.es/repository/> with the same name as in GitHub

## Software

### LAP Respository

NAME

<https://biocomputationlab.com/>

DEVELOPER

[www.laprepo.com](http://www.laprepo.com)

SOURCE  
LINK

#### Note

The App with version 6.3.1 analyzes your protocol before setting a robot to run, so the labware will not be shown before assigning the protocol to a specific robot when you import it into the App.

## 5.2 *Select Robot to Perform Script*

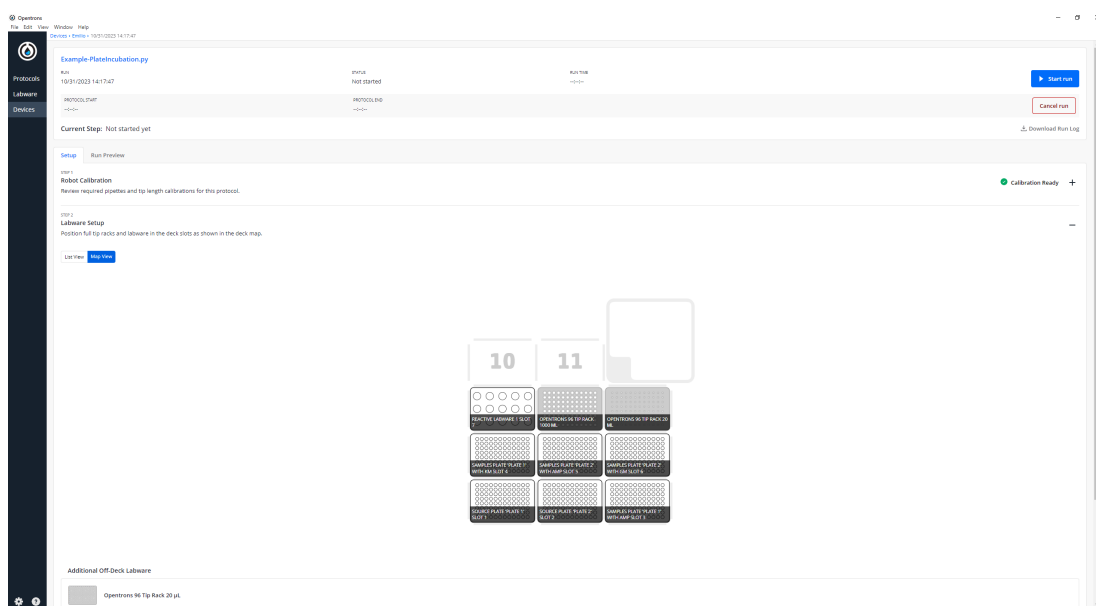
**Click in the protocol -> Start setup -> Choose the OT where the file *VariablesPlateIncubation.xlsx* is -> Proceed To Setup**

After clicking on Proceed to Setup, you should obtain the labware positions in the *Labware* tab and the reagents, with their corresponding volume, in the *Liquids* tab.

In case the protocol with the set variables cannot run, an error will occur during the run. Many errors are contemplated already and have a specific message that hints at what could have gone wrong.

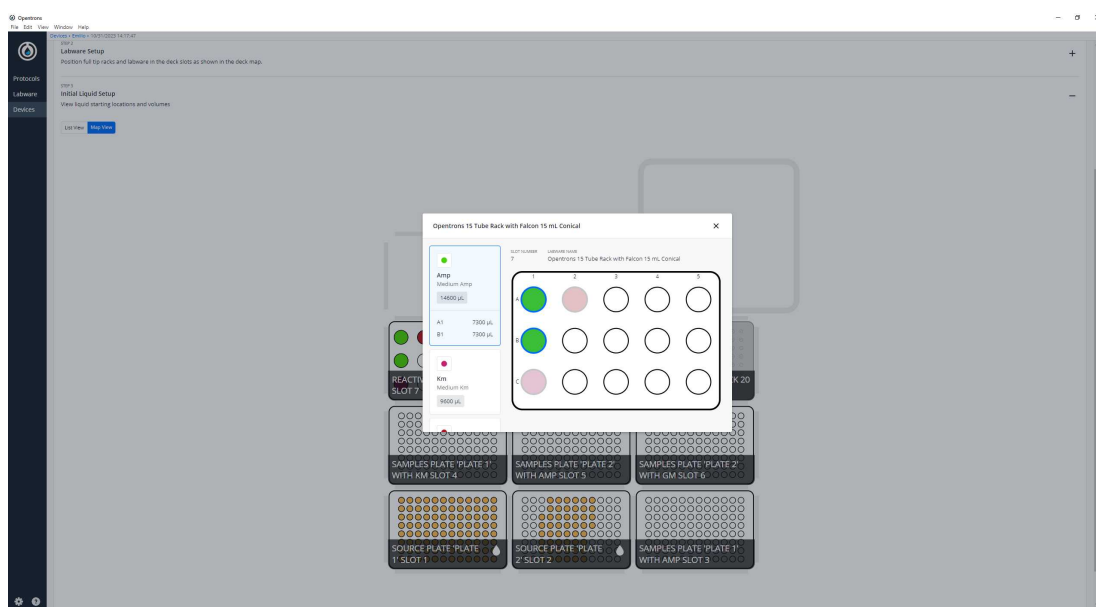
## Expected result

A labware setup should look like the following image, where you can find the initial and final plates, the Falcon labware to store the reagents and the corresponding tips



### Labware Set-up example of the Incubation Protocol

A liquid setup should look like the following image, where you can find the samples in the initial plates and the reactions in the 15-mL tube labware



### Liquid Set-up Example of the Incubation Protocol



#### Note

The volume of the initial samples is established to be 90% of the max volume of the well, but this is only a recommendation. **Just make sure that there is enough volume to transfer to all the final plates.**

On the other hand, the volume of the reagents is precisely what is needed, so it is **suggested always to add more to consider the pipetting error.**

#### Note

**It is recommended that you perform a labware position check.**

You can do it with test plates after loading the script but before cleaning the surface. That way, you reduce the probability of contamination (using the test plates and labware) and pipetting errors (position check).

## 6 Run Protocol in OT

### 6.1 *Make sure the needed calibrations are done*

Pipettes, tip racks and tip length calibrations need to be done for the items used in this run.

### 6.2 *Labware position check is performed (if needed)*



### 6.3 *Clean the surface of the robot with 70% ethanol to clean and disinfect the surfaces*

#### Note

Check the Opentrons page <https://support.opentrons.com/s/article/Cleaning-your-OT-2?> for more information about cleaning the OT-2 robot with the proper materials.

### 6.4 *Set the labware and reagents as shown in the OT-App*

### 6.5 *Start Run*

The procedure that the robot is going to do is mainly divided into 2 parts:

1. Distribute each reactive to the final plate(s) with the single-channel pipette
2. Distribute samples to the final plate(s) with the multi-channel pipette

#### Expected result

One or more plates with different reagents and same samples as set in the input variable file

**This set of plates will be given for each source plate the user has provided with the reagents the user has provided, as well**

## After-Running

### 7 Retrieve labware from the OT

## Example

1h

- 8 We want to transfer samples from 2 source plates, with 96 and 50 samples, to different antibiotic plates, one per antibiotic and source plate.  
The first one will be mixed with Ampicillin and Kanamycin. The second plate will be mixed with Ampicillin and Gentamycin.

We will use a computer with a Windows 10 system.

8.1 Excel template that we can find [⇒ go to step #1](#) filled with the name *VariablesPlateIncubation* 10m

 VariablesPlateIncubation.xlsx 17KB

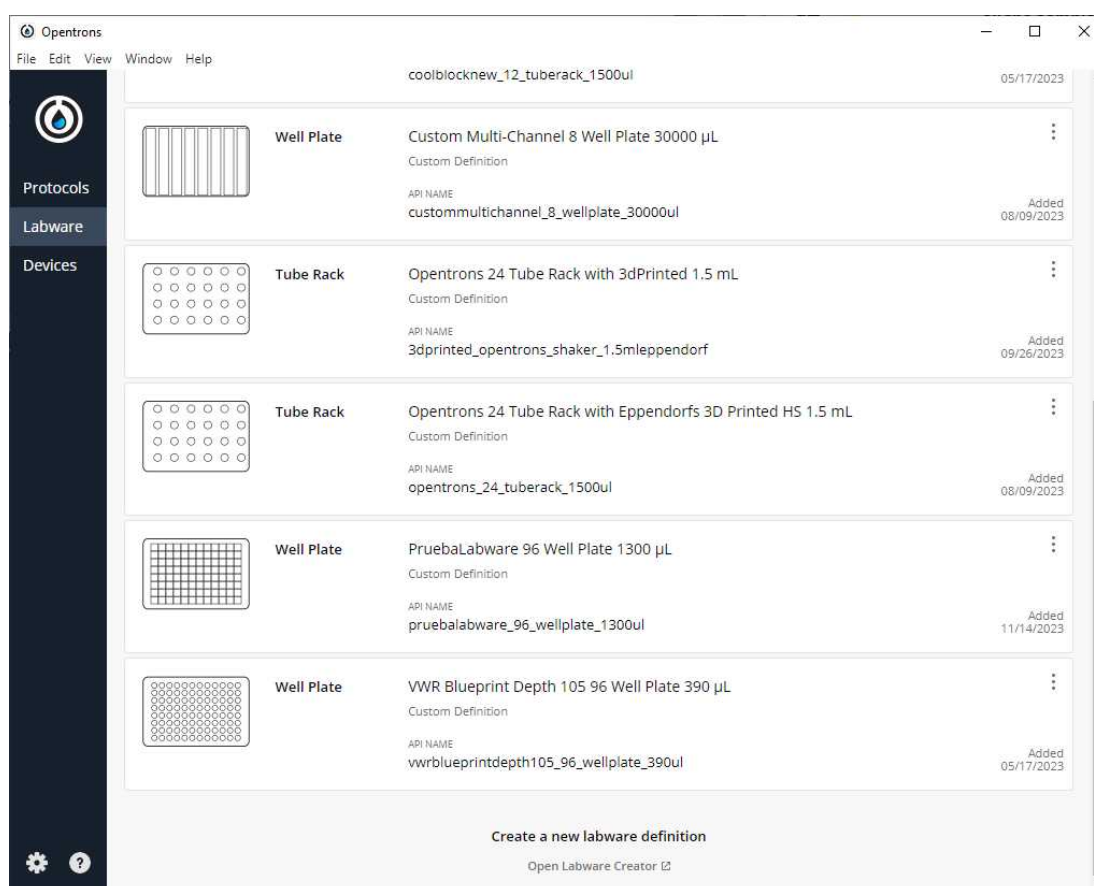
## 8.2 Upload custom labware to app

1m

We are using a custom labware called *vwrblueprintdepth105\_96\_wellplate\_390ul* that has been created with the labware creator that opentrons offers (<https://labware.opentrons.com/create/>)

 vwrblueprintdepth105\_96\_wellplate\_390ul.json 11KB

Upload it to the labware and ensure it is loaded in the app.



List of custom labware recorded in the Opentrons App

8.3 Because we are using version 1.0.0 of the script in this example, we will transfer the directory of the labware as well (here we have attached a zip, but it is the folder that must be transferred, not the zip) 1m

 vwrblueprintdepth105\_96\_wellplate\_390ul.zip 1KB

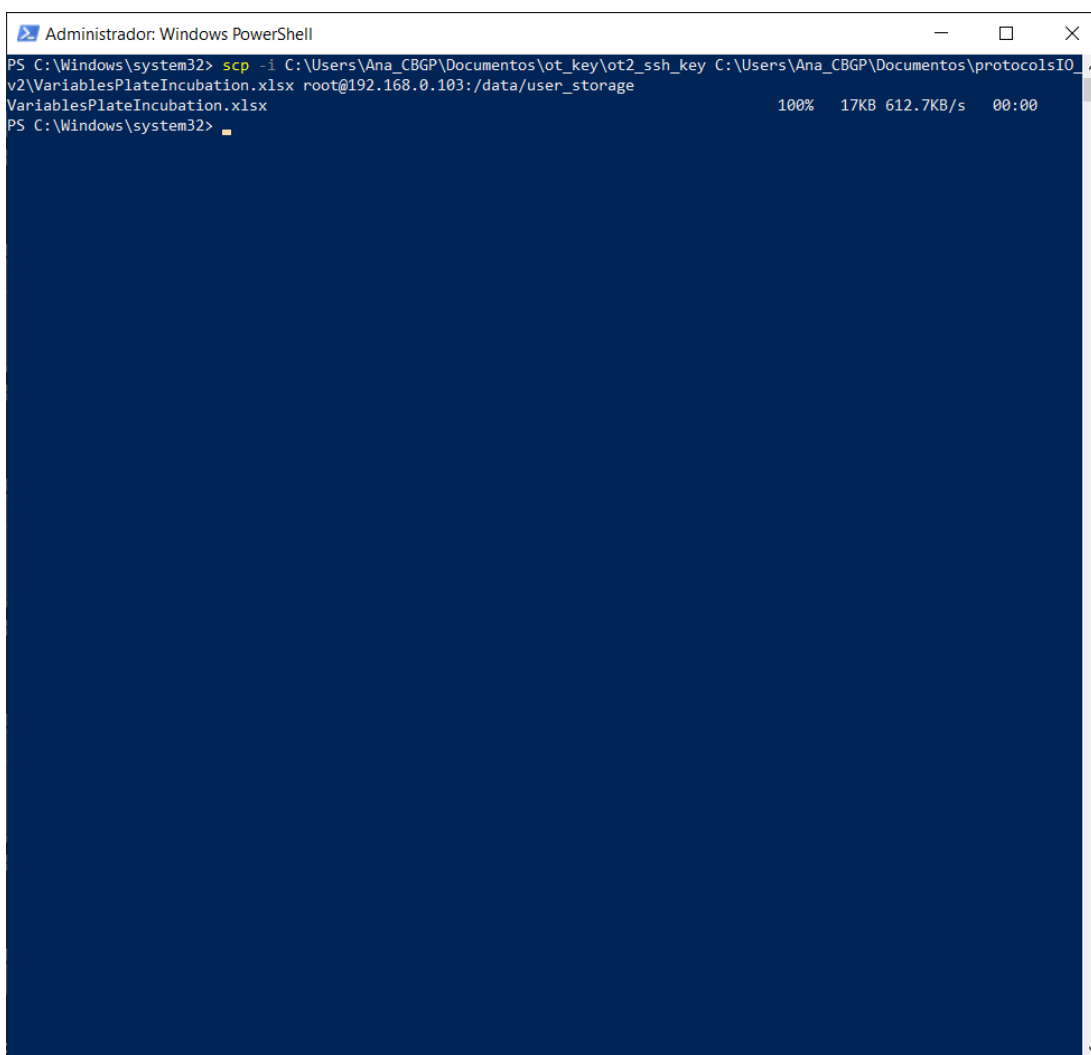
## Command

### Transferring the used custom labware to OT (Linux)

```
scp -i [ot_key] -r vwrblueprintdepth105_96_wellplate_390ul root@[IP_OT]:/data/labware/v2/custom_definitions/custom_beta
```

## 8.4 Export the variable file to *the* /data/user\_storage folder in the robot.

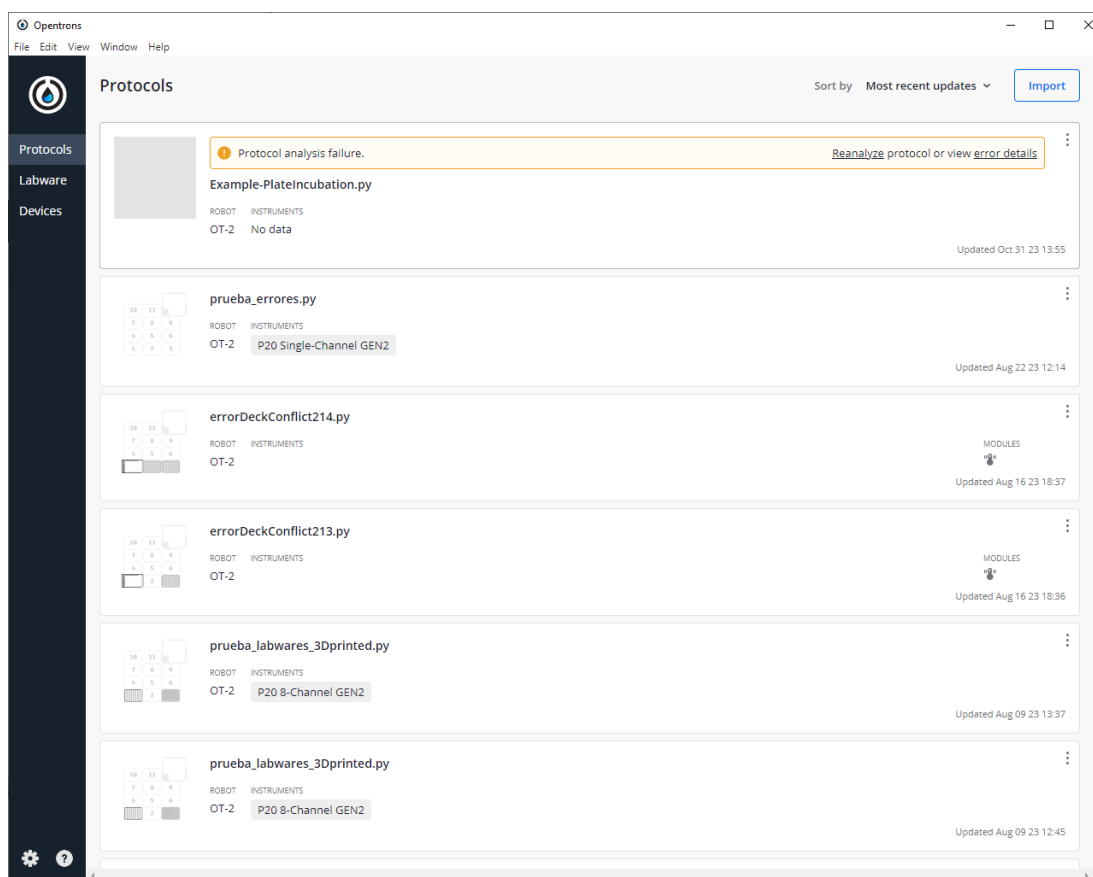
2m



command line window with scp commands to transfer the variables .xlsx from our computer to the OT-2

8.5 Import the script that we have downloaded from [Pythonade](#) from [go to step #5.1](#) (I named it *Example-PlateIncubation.py*) to the OT-App 30s

 Example-PlateIncubation.py 45KB



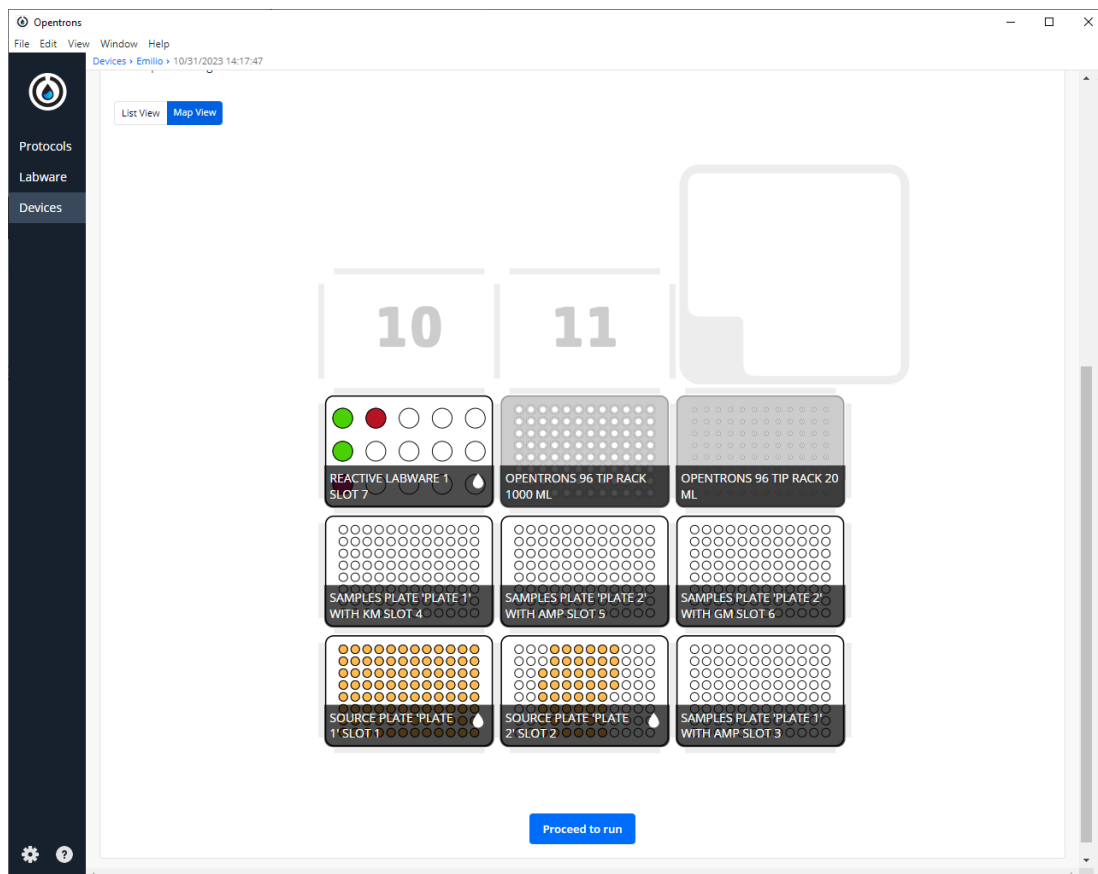
Result of importing the Python script in the OT-App

As we can see, we have an error, but that is programmed because the script is meant to work in the robot but not in your computer

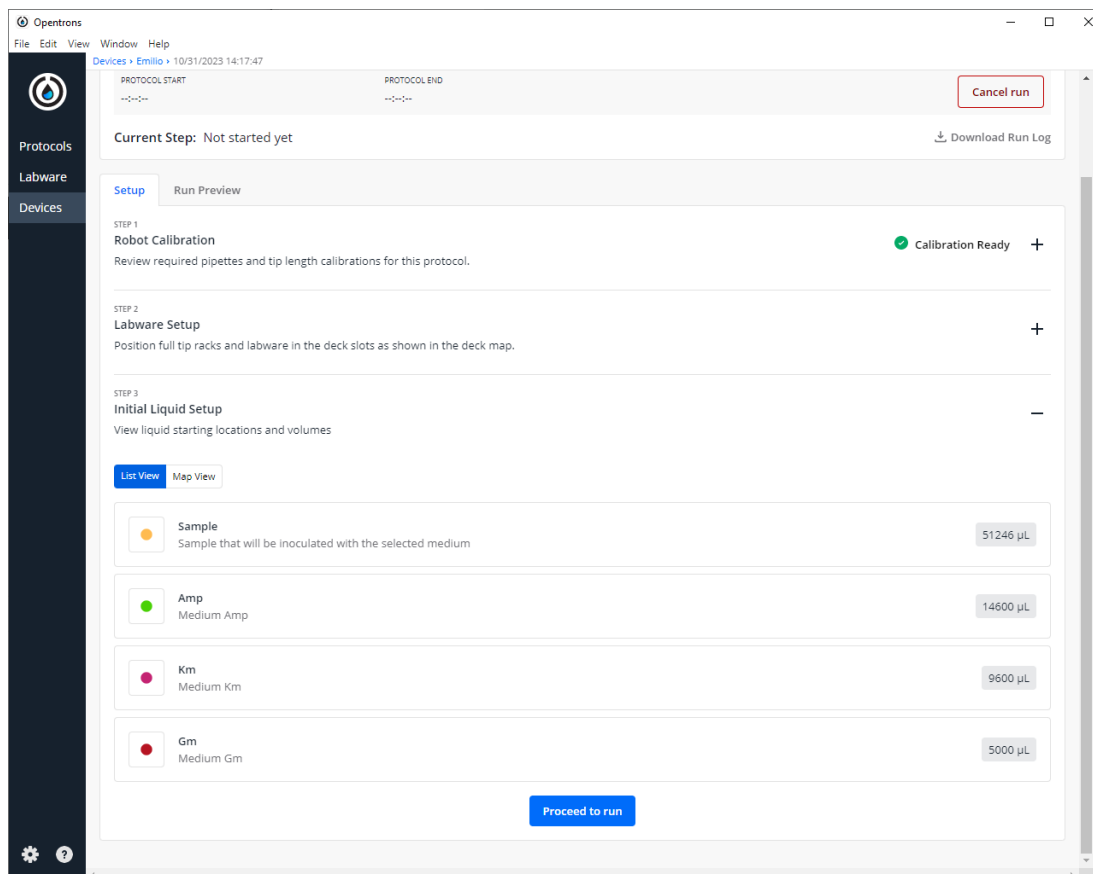
8.6 Run the protocol in the robot that we have transferred the Excel file 1m

**Example-PlateIncubation.py** -> **Start setup** -> **Select** the **robot** in which we are going to run the protocol

If we do not have any errors, the output should look like the following pictures.



Labware and liquid set-up layout



Volumes of the antibiotics needed to perform the protocol

8.7 Turn the HEPA filter module

30s



8.8 Clean platform of the robot that we are going to perform the protocol

2m

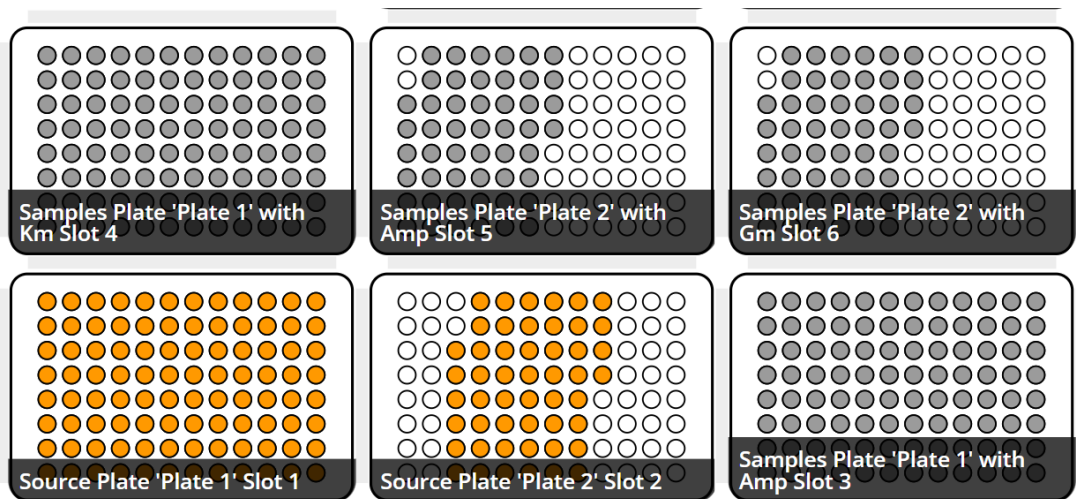
8.9 Prepare all reagents and labware in the places the App is showing and take into account the notes  
step [go to step #5.2 Notes](#)

5m

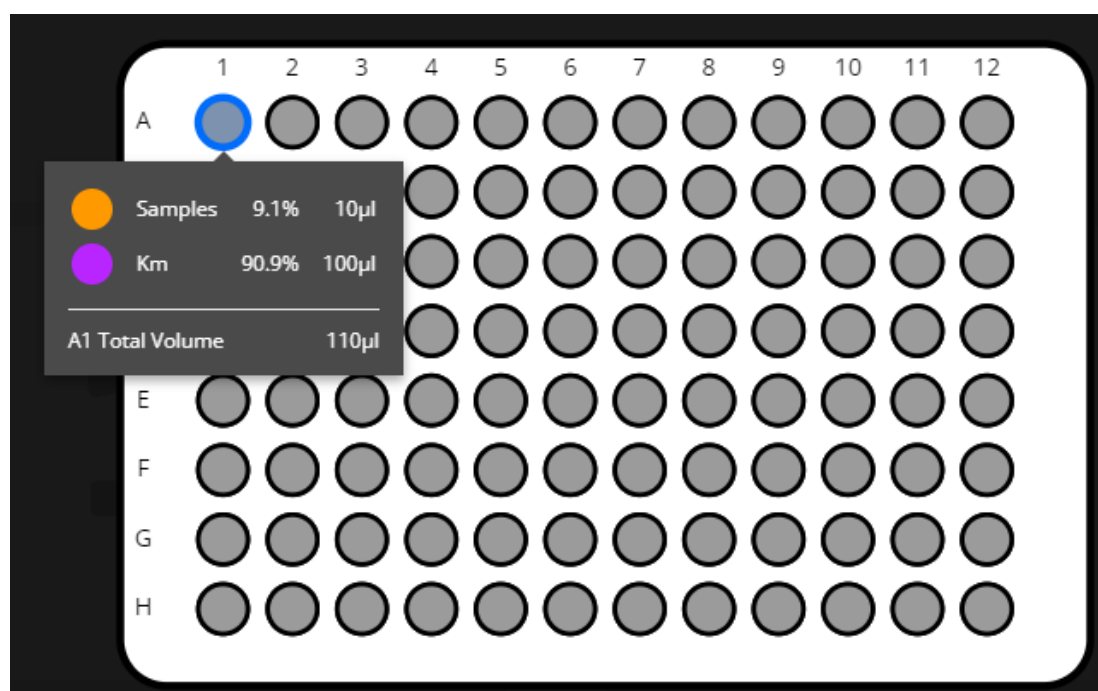
8.10 Start Run

32m

## Expected result



Final labware layout in which the grey wells are a mix of the correspondent reactive with the sample



Example of the content of A1 in the labware *Sample Plate 'Plate 1' with Km Slot 4*

Here, we will obtain the mix between the volume of media and the samples set in the variable file in the final plates. These positions are seen in the first image by the grey wells, and we can see the info on the plate and the media in the name of the labware and an example of the composition of 1 well in the second picture.



## 8.11 Retrieve labwares from the OT

5m