Decentralized Differential Gene Analysis

1. Introduction
2. Method
    2.1 Requirements
    2.2 Create Deployment
    2.3 Connect to the deployed VM using SSH
    2.4 Mount Storj to the deployed VM
3. Other Tips:

1. Introduction

High performance computation is required to process next generation sequencing data and translate experimental findings from biological experiments to actionable insights. There are many benefits to using decentralized physical infrastructure networks (DePINs) for this purpose. Unlike cloud service providers that requires operational costs associated with running large scale data centers and have to pay increasingly expensive insurance premiums to insure against the weather volatility associated with climate change, DePINs hosted by smaller scale providers can offer globally competitive prices. Here I present a technical protocol that enables permissionless, borderless processing of biological data for clinical research. This technical documentation is a demonstration of a basic Nextflow workflow using Storj and Akash Network for decentralized data storage and computing services.

`nf-core` is a open source software community offering many genomics pipelines developed by the global scientific community. As a demo use case, I will use `nf-core/differentialabundance` to perform a differential gene analysis using only DePINs. I choose this pipeline because it has low hardware requirements to run and is sufficient to demonstration that if my protocol works, then any other genomics pipeline developed by the `nf-core` community should work too.

The aim of a differential gene analysis is to identify genes with difference in expression patterns under different experimental conditions (gene(s) versus condition(s) and vice versa) for a set of samples. This is a key step to connect RNA sequence data with experimental conditions and translate the biological experiment into findings that can help us map changes to **pathways, processes, and regulatory mechanisms to** generate hypotheses about the underlying biological mechanisms. Many clinical trials and clinical experiments using non-human model organisms depend on processing RNA data to understand the genetic mechanisms underlying how treatments affect individuals.

2. Method

2.1 Requirements

Token Addresses:

- STORJ (ER-20 Token address) [1] : 0xB64ef51C888972c908CFacf59B47C1AfBC0Ab8aC
- AKT (Akash Network) [2]: akash19hpqkecf674zhqc33j5f2dnezjp636epshzrgz

Browser Wallet: Kepler [3]

Hardware Requirements:

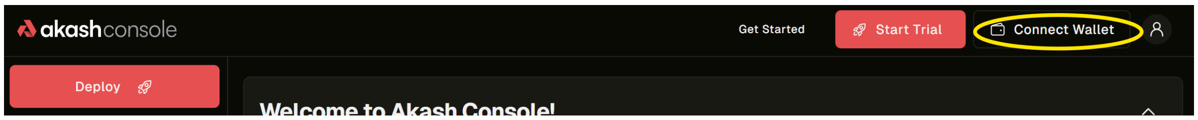
- 8 CPU
- No GPU
- 14 GB memory
- Ephemeral storage: 20 GB

System Requirements: Ubuntu 24.4

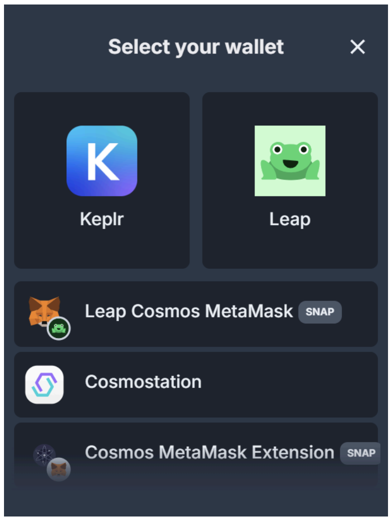
Dataset: https://link.storjshare.io/s/jwwea5lbiwqnnetf3uqbk2ptp7uq/test

2.2 Create Deployment

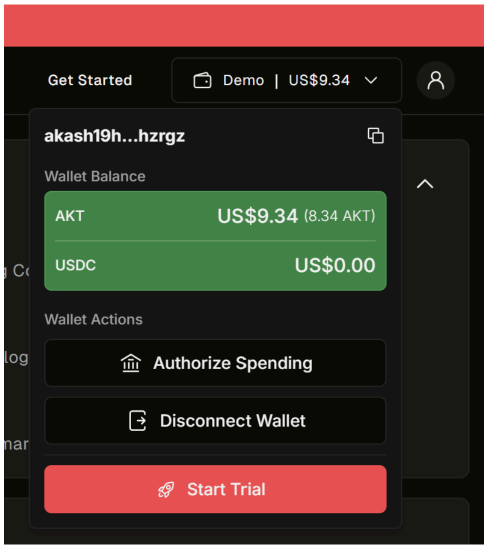
1. Go to https://console.akash.network/, click on “Connect Wallet.”



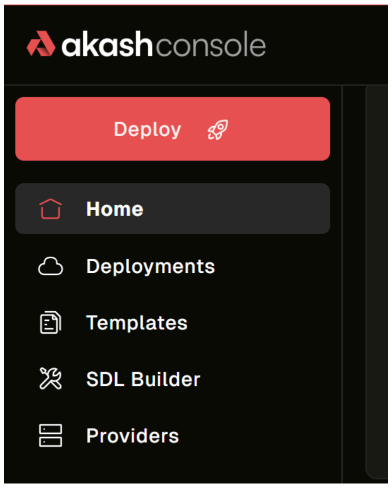
2. Select a wallet that contains your AKT tokens. I recommend Keplr.

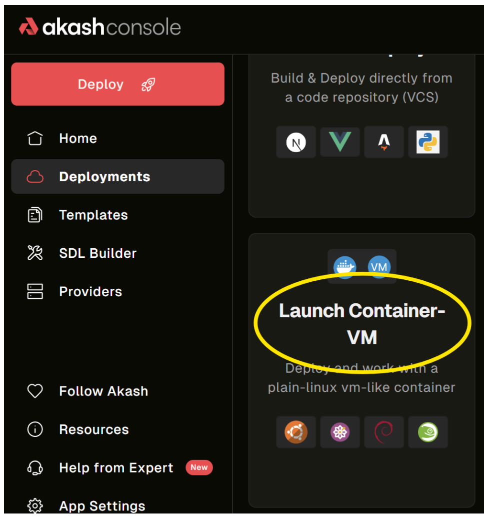


3. If the wallet connect correctly, hover the cursor over the wallet drop down menu and you will see AKT tokens in the connected wallet:



4. Click on “Deploy” > “Launch Container-VM”

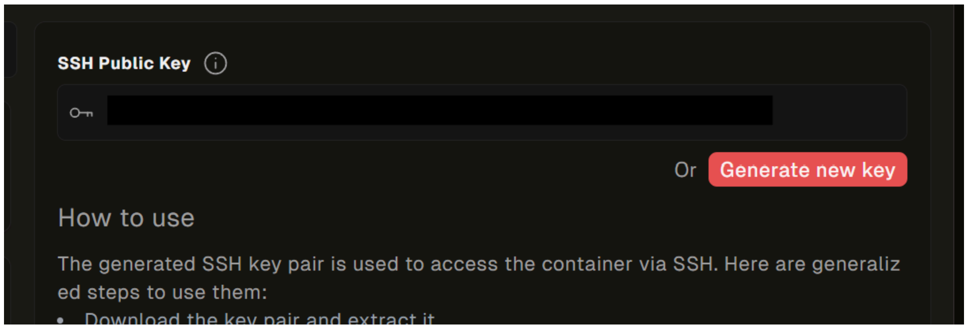




Build & Deploy directly from a code repository (VCS)

Deploy and work with a plain-linux vm-like container

5. Select “Ubuntu 24.04” and fill in the hardware requirements for running the analysis (2.1 Requirements).
6. Generate the SSH Public Key.



SSH Public Key

[redacted]

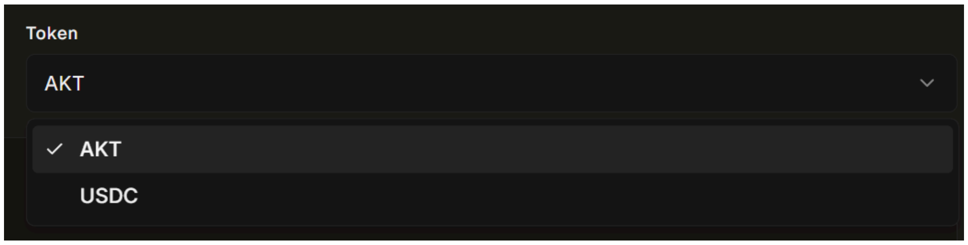
Or Generate new key

How to use

The generated SSH key pair is used to access the container via SSH. Here are generalized steps to use them:

- Download the key pair and extract it

7. Scroll to the bottom to select payment currency in AKT or USDC.



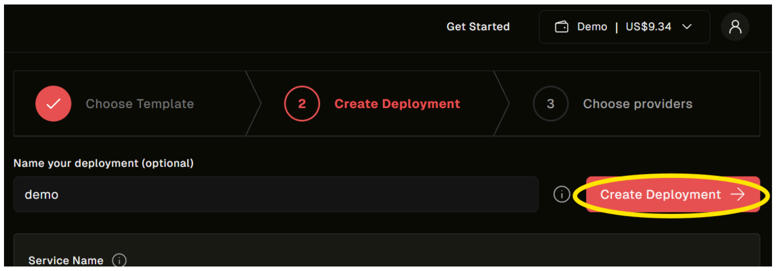
Token

AKT

✓ AKT

USDC

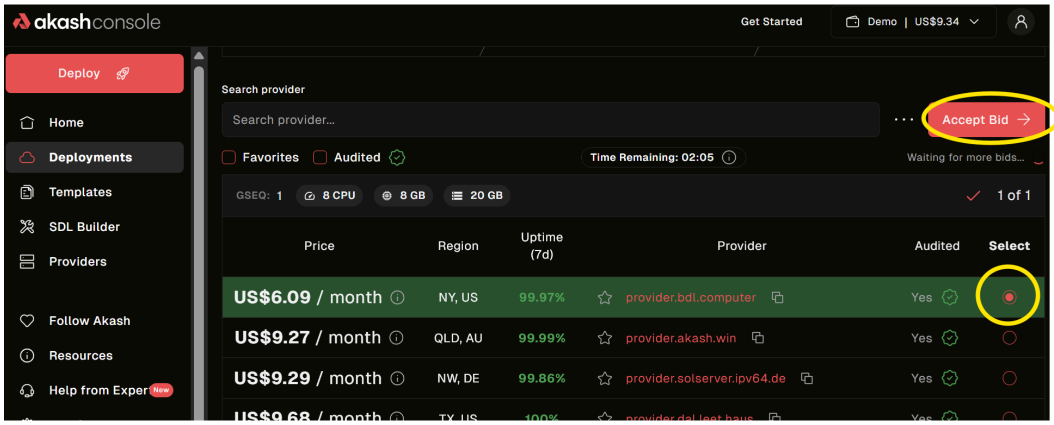
8. Select “Create Deployment.”



Name your deployment (optional)

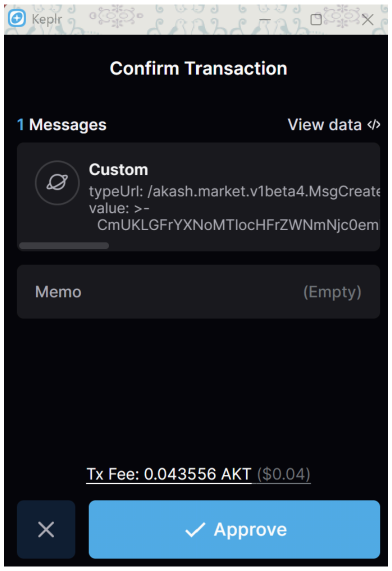
demo

9. Click on “Continue” after confirming the deployment requirements.
10. Approve the transaction request in the browser wallet pop-up.
11. Select a service provider > Click on “Accept Bid”



Price	Region	Uptime (7d)	Provider	Audited	Select
US$6.09 / month	NY, US	99.97%	provider.bdl.computer	Yes	●
US$9.27 / month	QLD, AU	99.99%	provider.akash.win	Yes	○
US$9.29 / month	NW, DE	99.86%	provider.solserver.ipv64.de	Yes	○
US$9.68 / month	TX, US	100%	provider.dal.leet.haus	Yes	○

12. Approve the transaction request.


Keplr

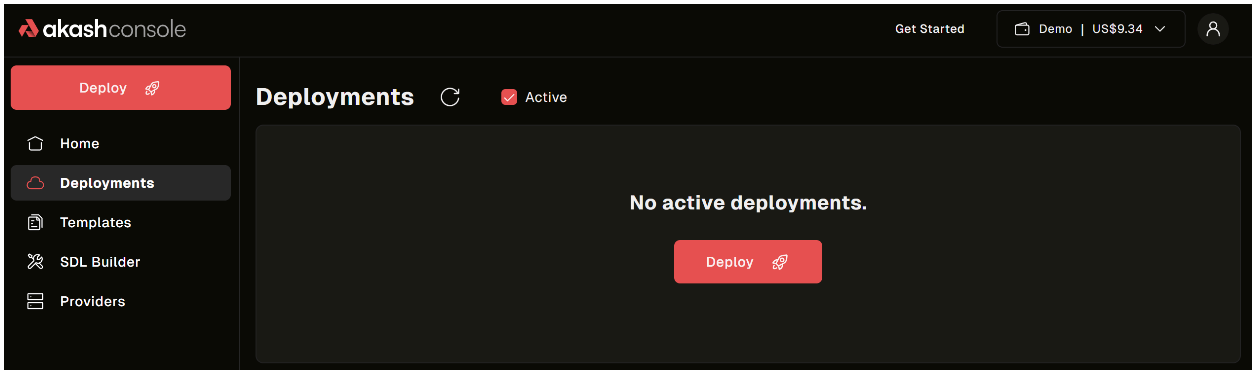
Confirm Transaction

1 Messages View data

Custom
typeUrl: /akash.market.v1beta4.MsgCreate
value: >-
 CmUKLGFrYXNoMTlocHFrZWNmNjc0em

Memo (Empty)

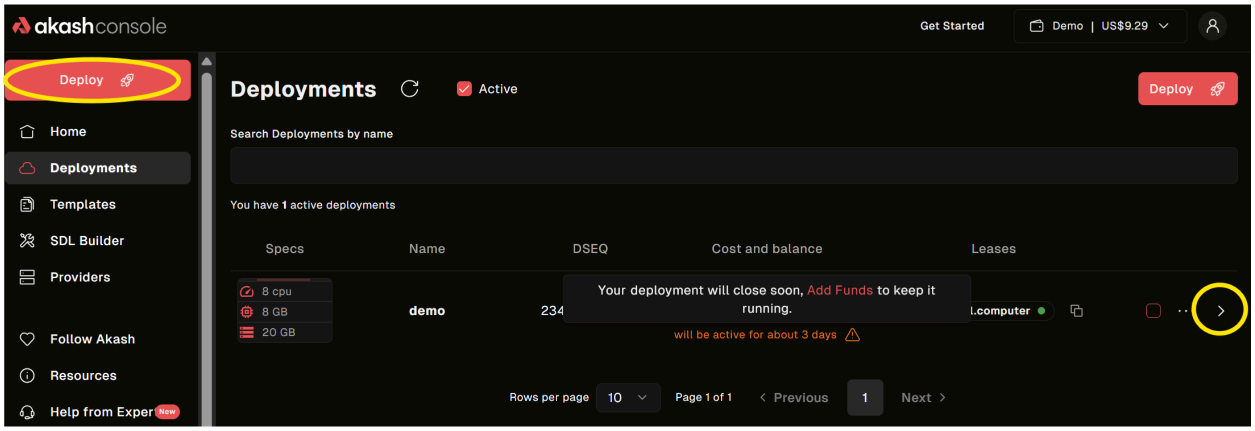
Tx Fee: 0.043556 AKT ($0.04)



No active deployments.

2.3 Connect to the deployed VM using SSH

1. Go to "Deployments" then select ">"



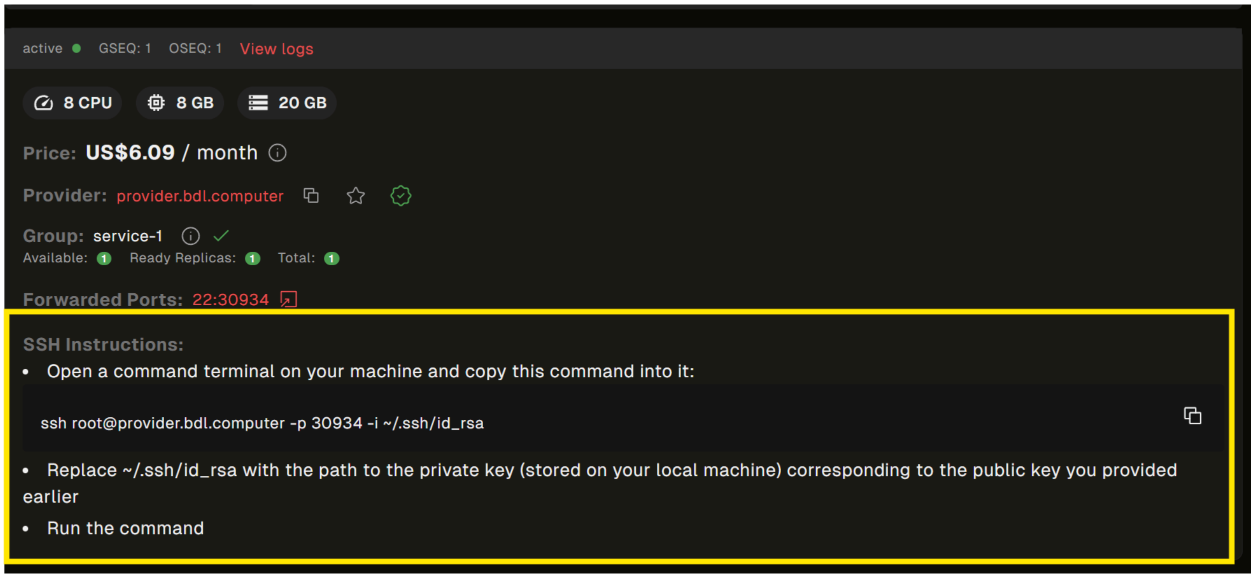
Search Deployments by name

You have 1 active deployments

Specs	Name	DSEQ	Cost and balance	Leases
8 cpu 8 GB 20 GB	demo	234	Your deployment will close soon, Add Funds to keep it running. will be active for about 3 days	.computer

Rows per page 10 Page 1 of 1 Previous 1 Next

2. Follow the SSH instructions.



Price: **US$6.09** / month

Provider: provider.bdl.computer

Group: service-1
Available: 1 Ready Replicas: 1 Total: 1

Forwarded Ports: 22:30934

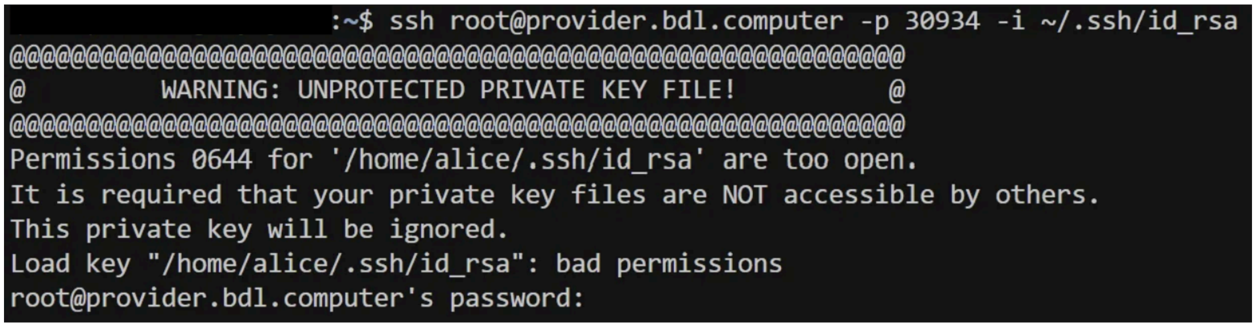
SSH Instructions:

- Open a command terminal on your machine and copy this command into it:

```
ssh root@provider.bdl.computer -p 30934 -i ~/.ssh/id_rsa
```

- Replace ~/.ssh/id_rsa with the path to the private key (stored on your local machine) corresponding to the public key you provided earlier
- Run the command

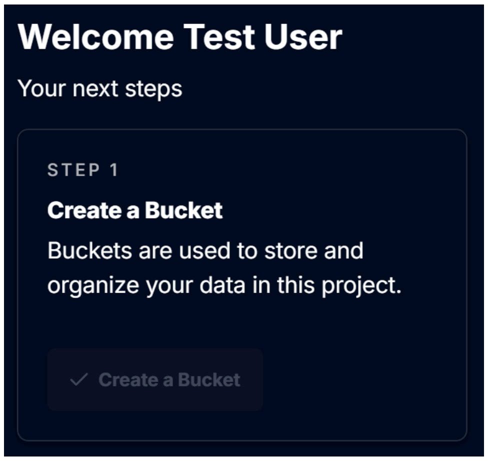
3. If you see the error below , run `chmod 600 ~/.ssh/id_rsa`



```
[redacted]:~$ ssh root@provider.bdl.computer -p 30934 -i ~/.ssh/id_rsa
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for '/home/alice/.ssh/id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/home/alice/.ssh/id_rsa": bad permissions
root@provider.bdl.computer's password:
```

2.4 Mount Storj to the deployed VM

1. After creating a Storj account, create a new bucket.


Welcome Test User

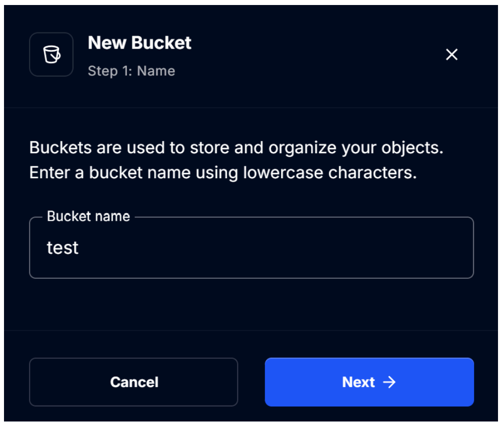
Your next steps

STEP 1

Create a Bucket

Buckets are used to store and organize your data in this project.

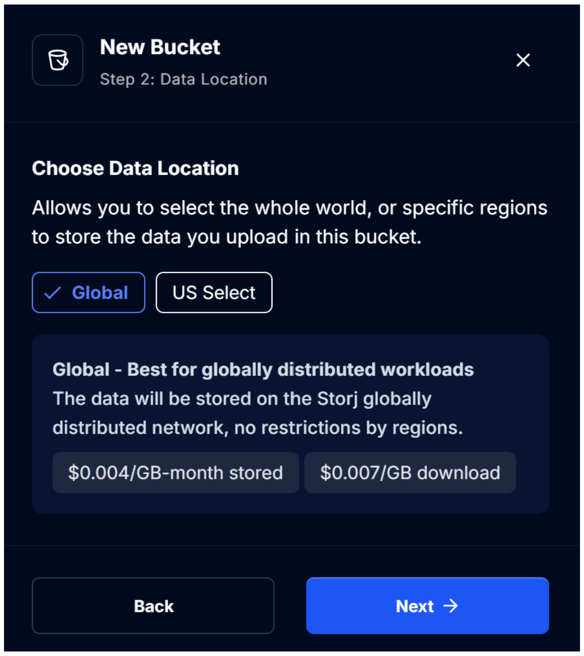
2. Follow through the prompted steps to create a new bucket. For demo purposes, I selected “No” in Step 3. For reproducible results, select “Yes” and determine the best option for the use case.


New Bucket

Step 1: Name

Buckets are used to store and organize your objects. Enter a bucket name using lowercase characters.

Bucket name: test


New Bucket

Step 2: Data Location

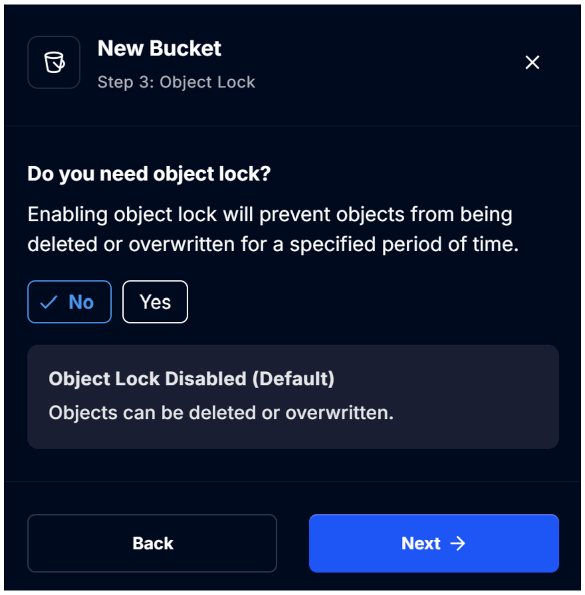
Choose Data Location

Allows you to select the whole world, or specific regions to store the data you upload in this bucket.

[✓ Global] US Select

Global - Best for globally distributed workloads
The data will be stored on the Storj globally distributed network, no restrictions by regions.

$0.004/GB-month stored $0.007/GB download

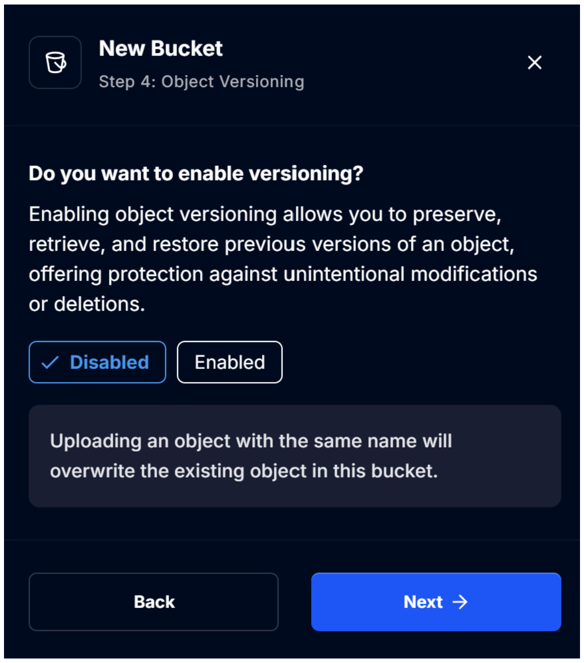

New Bucket
Step 3: Object Lock

Do you need object lock?

Enabling object lock will prevent objects from being deleted or overwritten for a specified period of time.

[✓ No] Yes

Object Lock Disabled (Default)
Objects can be deleted or overwritten.


New Bucket
Step 4: Object Versioning

Do you want to enable versioning?

Enabling object versioning allows you to preserve, retrieve, and restore previous versions of an object, offering protection against unintentional modifications or deletions.

[✓ Disabled] Enabled

Uploading an object with the same name will overwrite the existing object in this bucket.

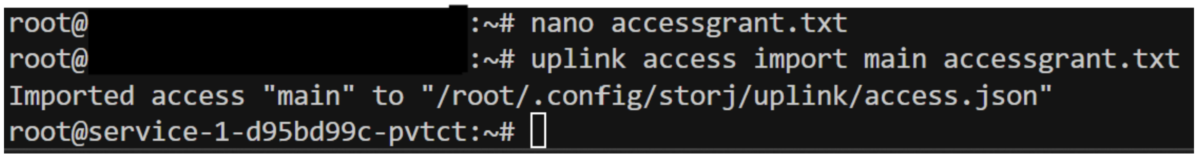
3. Install Storj Uplink to download and upload files [4].

```
apt update
apt-get install curl unzip nano sudo wget
curl -L https://github.com/storj/storj/releases/latest/download/uplink_linux
_amd64.zip -o uplink_linux_amd64.zip
unzip -o uplink_linux_amd64.zip
install uplink /usr/local/bin/uplink
```

4. Follow the instructions here to create an Access Grant [5].
5. Return to the Akash VM and run the following:

```
touch accessgrant.txt
nano accessgrant.txt
```

6. Copy and paste Access Grant key to `accessgrant.txt` .
7. Follow the instructions here to set up Uplink CLI with the Access Grant [6]. If the setup was successful, you will see the following:



```
root@[redacted]:~# nano accessgrant.txt
root@:~# uplink access import main accessgrant.txt
Imported access "main" to "/root/.config/storj/uplink/access.json"
root@service-1-d95bd99c-pvtct:~# 
```

8. To view the listed bucket, run `uplink ls sj://test`
9. I already set up the folder with the reference genomes files and Nextflow input files required to run `nf-core/differentialabundance`. Download the folders, `data` and `reference_genome` :

```
# Initialize directories
mkdir ~/data
mkdir ~/reference_genome/dmel/

# Copy the files from Storj to Akash VM
uplink cp sj://test/data/salmon.merged.gene_counts.tsv ~/data/
uplink cp sj://test/data/salmon.merged.gene_lengths.tsv ~/data/
uplink cp sj://test/data/new_samplesheet.csv ~/data/
uplink cp sj://test/data/contrasts.csv ~/data/
uplink cp sj://test/reference_genome/Drosophila_melanogaster.BDGP6.46.113.gtf.gz ~/reference_genome/dmel/
uplink cp sj://test/reference_genome/GO_All_Drosophila_melanogaster_GeneSymbol.gmt ~/reference_genome/dmel/
uplink cp sj://test/reference_genome/Pathways_Reactome_Drosophila_melanogaster_GeneSymbol.gmt ~/reference_genome/dmel/

cd ~/reference_genome/dmel/
gzip -dk Drosophila_melanogaster.BDGP6.46.113.gtf.gz
```

2.4 Running Nextflow on Akask VM

1. Install JAVA and Nextflow:

```
cd ~/

# Install Java
```

```
sudo apt install default\-jre

# Set Java memory limit
# Add to .bashrc directly
echo 'NXF_OPTS="-Xms1g -Xmx4g"' >> ~/.bashrc

# Source .bashrc to apply changes
source ~/.bashrc
```

2. Install Nextflow:

```
# Install Nextflow
curl -s https://get.nextflow.io | bash
```

3. Install Conda:

```
# Install conda
# Download the Miniconda installer
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_
64.sh

# Make it executable
chmod +x Miniconda3-latest-Linux-x86_64.sh

# Run the installer
bash Miniconda3-latest-Linux-x86_64.sh

conda update conda

# Follow the prompts and agree to the license terms
# Choose installation location (default is usually fine)
# Allow conda init when prompted (recommended)
source ~/.bashrc
```

2. Run the command below to run `nf-core/differentialabundance` using the downloaded files:

```
./nextflow run nf-core/differentialabundance \
--input ~/data/new_samplesheet.csv \
--contrasts ~/data/contrasts.csv \
--matrix ~/data/salmon.merged.gene_counts.tsv \
--transcript_length_matrix ~/data/salmon.merged.gene_lengths.tsv \
--gtf /root/reference_genome/dmel/Drosophila_melanogaster.BDGP6.46.11
3.gtf \
--gene_sets_files /root/reference_genome/dmel/GO_All_Drosophila_melan
ogaster_GeneSymbol.gmt,\
/root/reference_genome/dmel/Pathways_Reactome_Drosophila_melanoga
ster_GeneSymbol.gmt \
--outdir ~/scratch/dge_analysis_filtered \
-w ~/scratch/work/dge_analysis \
-profile rnaseq,conda \
--gprofiler2_run true \
--gprofiler2_organism dmelanogaster \
--gprofiler2_sources "GO,GO:MF,GO:BP,GO:CC,KEGG,REAC" \
--gprofiler2_correction_method gSCS \
--filtering_min_proportion 0.3 \
--filtering_grouping_var condition \
--shinyngs_build_app false
```

- To learn what this means, please join my community on Nas.io: https://nas.io/degenseq

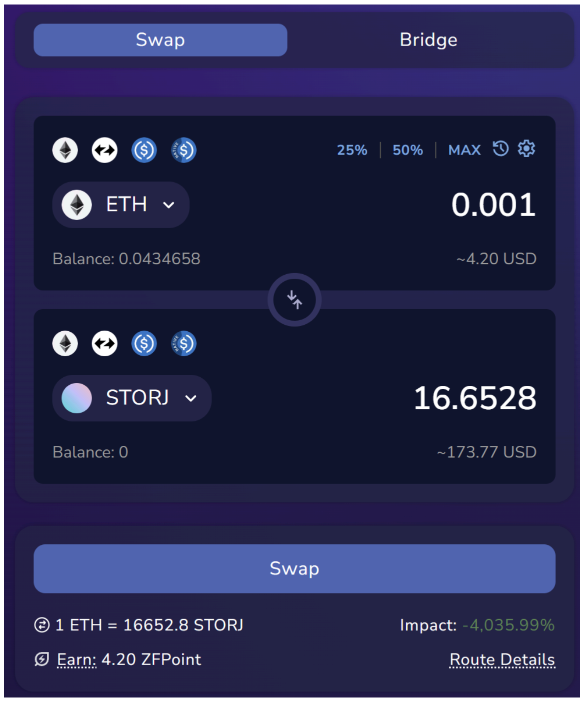
3. Compress the output folder and upload the output files to Storj.

```
# compress the output folder
cd ~/scratch
tar -zcvf dge_analysis_filtered.tar.gz dge_analysis_filtered
uplink cp ~/scratch/dge_analysis_filtered.tar.gz sj://test/dge_analysis_filter
ed.tar.gz
```

4. To download/view the dataset used in this tutorial, please visit here.

3. Other Tips:

- Don’t use zksync Bridge [4] to bridge ETH from zksync to Ethereum network. It takes 6 hrs. Use Ctrl Wallet’s internal swap feature instead.
- There’s strong STORJ/ETH liquidity pool asymmetry on Zkswap finance [5]. Storj accepts STORJ token payments through the zksync and ETH networks. Why?



ETH 0.001
Balance: 0.0434658 ~4.20 USD

STORJ 16.6528
Balance: 0 ~173.77 USD

1 ETH = 16652.8 STORJ Impact: -4,035.99%
Earn: 4.20 ZFPoint Route Details