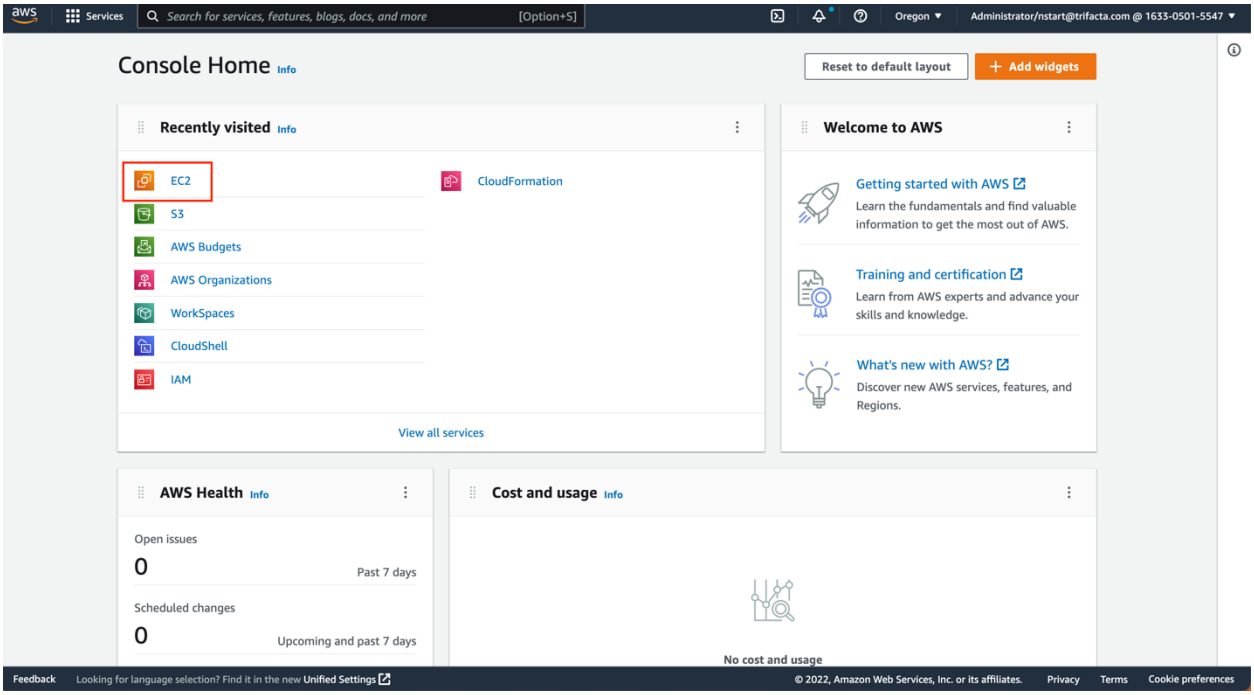
DESIGNER CLOUD
Powered by TRIFACTA

aws

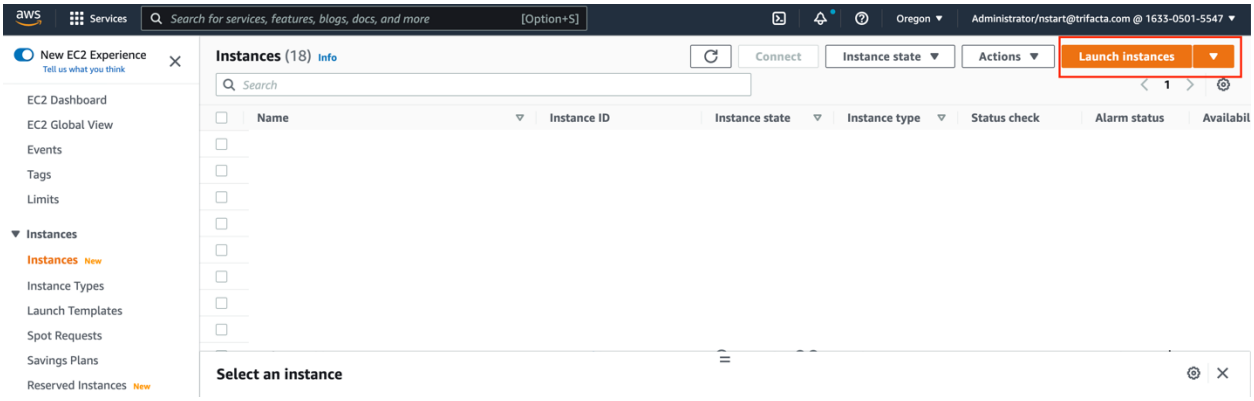
Objective

This document is to be used to automate a Designer Cloud v9.2 Customer Managed deployment in AWS on RHEL 8.6 using Ansible.

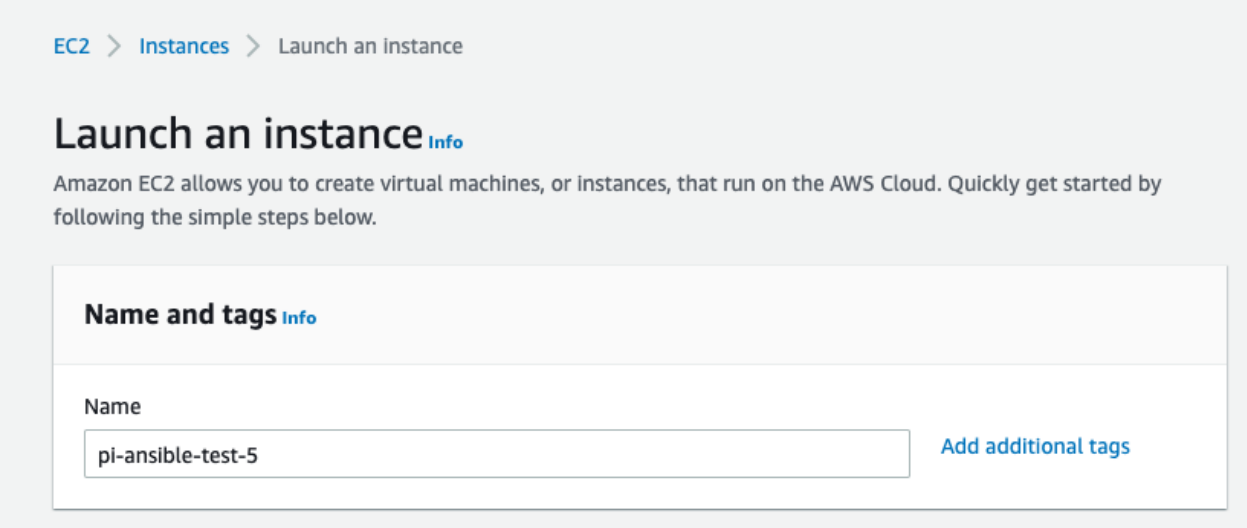
1: Login to AWS Console -> go to EC2



2. Click “Launch an instance”

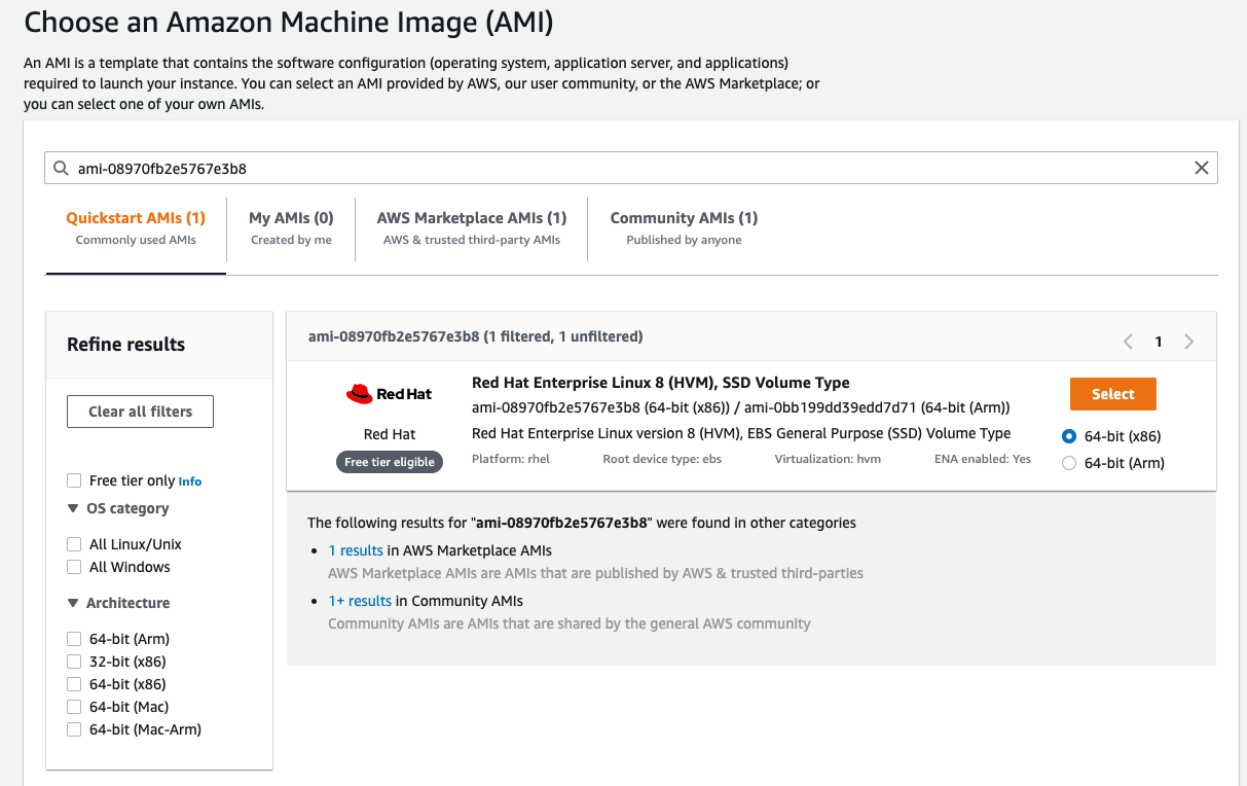


3. Under Names and Tags – add a name for your EC2 machine

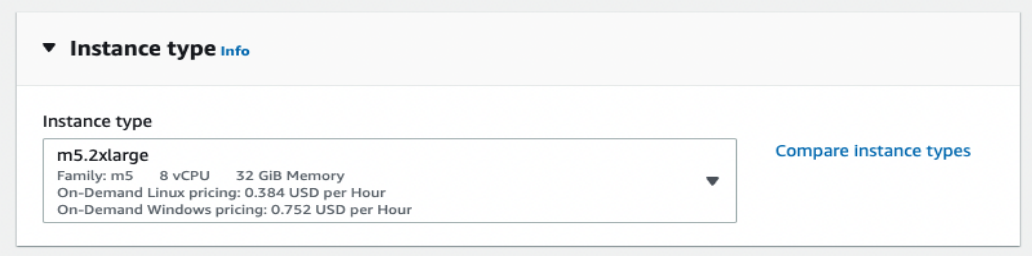


4. Search for AMI: ami-08970fb2e5767e3b8

In the Application and OS Images Amazon Machine Image (AMI). As you can see in the screenshot below, this is for RHEL 8.6.

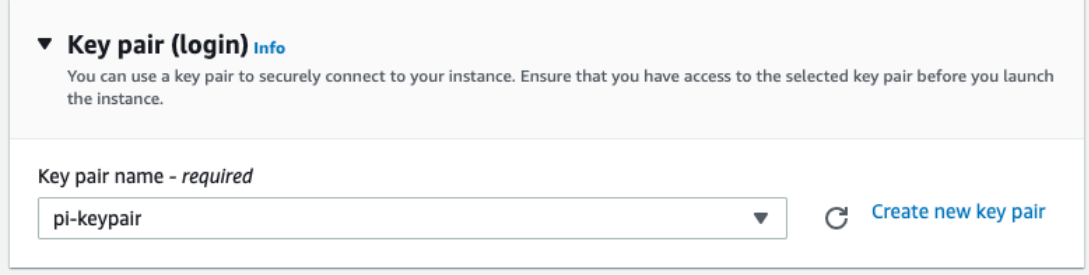


5. Pick instance type (recommend using m5.2xlarge)



6. Use an existing keypair or create one.

This will be used to SSH into the instance as well as connect to the instance from Ansible.

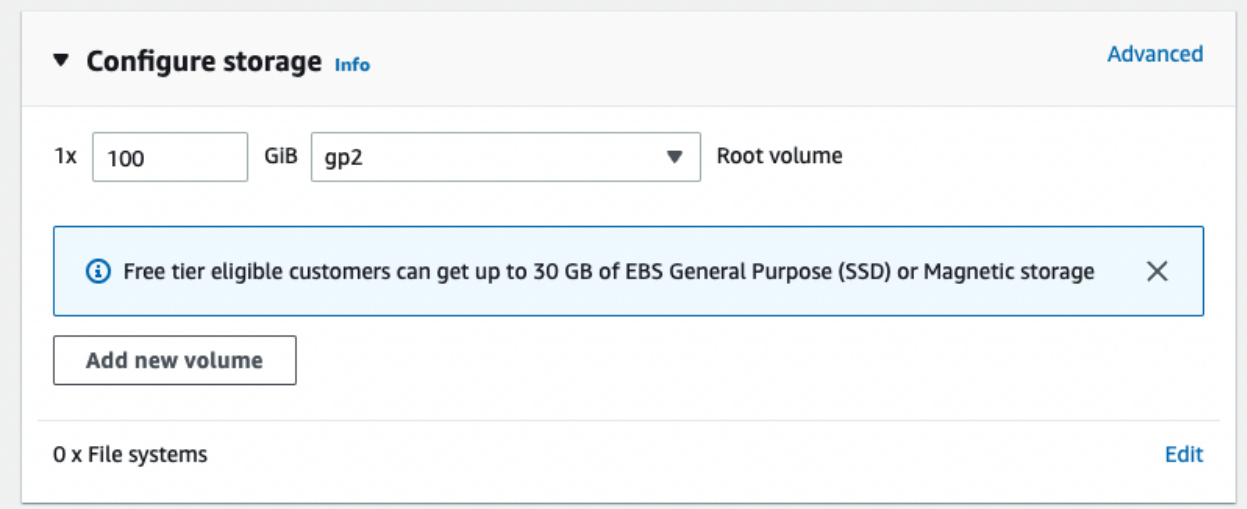


7. Network

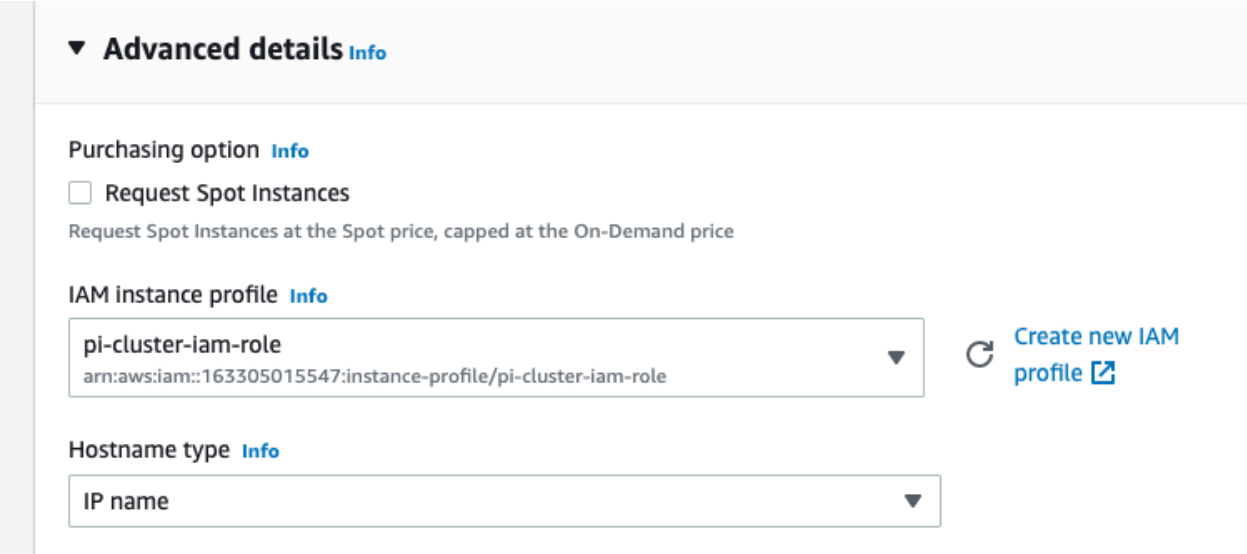
1. Select your VPC – (ie; vpc-028790bed3b4c1e33 Engineering)
2. Subnet needs to be public for the EC2 instance to get a public IP address. (For testing, this will auto-assign the correct subnet based on choosing your VPC).
3. Security Group (virtual firewall) should be configured such that the instance can be reached on port 22 (SSH) and 3005 (TCP HTTP for the UI).
   *(**May need to create a new Security Group (SG))*

8. Configure Storage

Change value to 1x 100 GIB



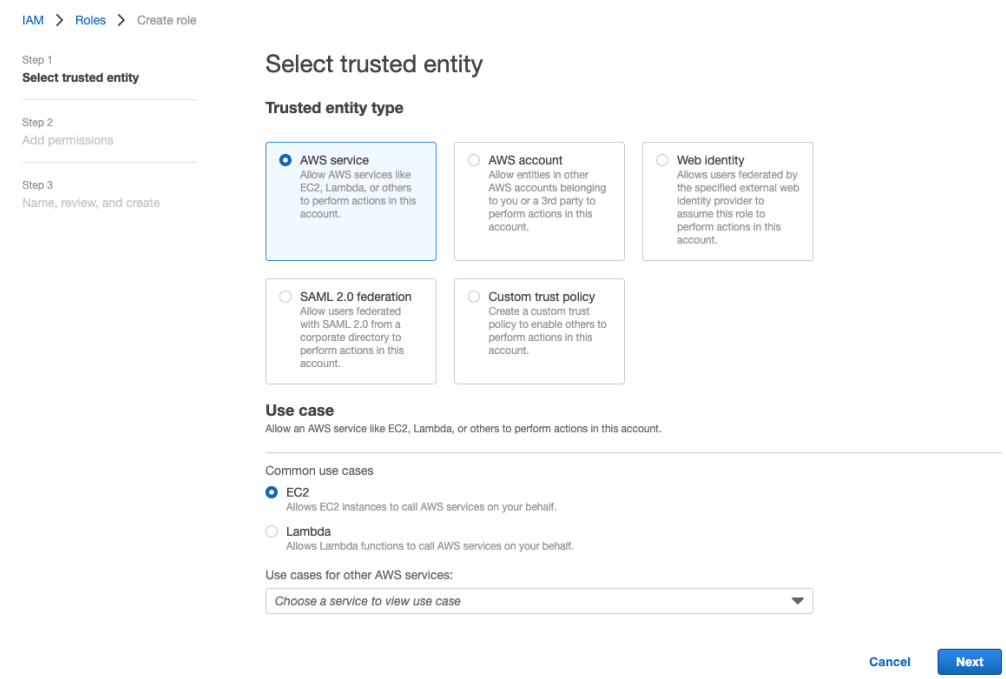
9. Click on **"Advanced details"** to expand this section. (Admin may need to select from an existing IAM instance profile.)



If you create a new IAM profile: (ie; Selection of an existing one in the screenshot above – pi-cluster-iam-role)

Note: If you don't create an instance role upon launching the EC2 instance, ***you can't add one later***. You can however modify the attached role and associated policies after the fact.

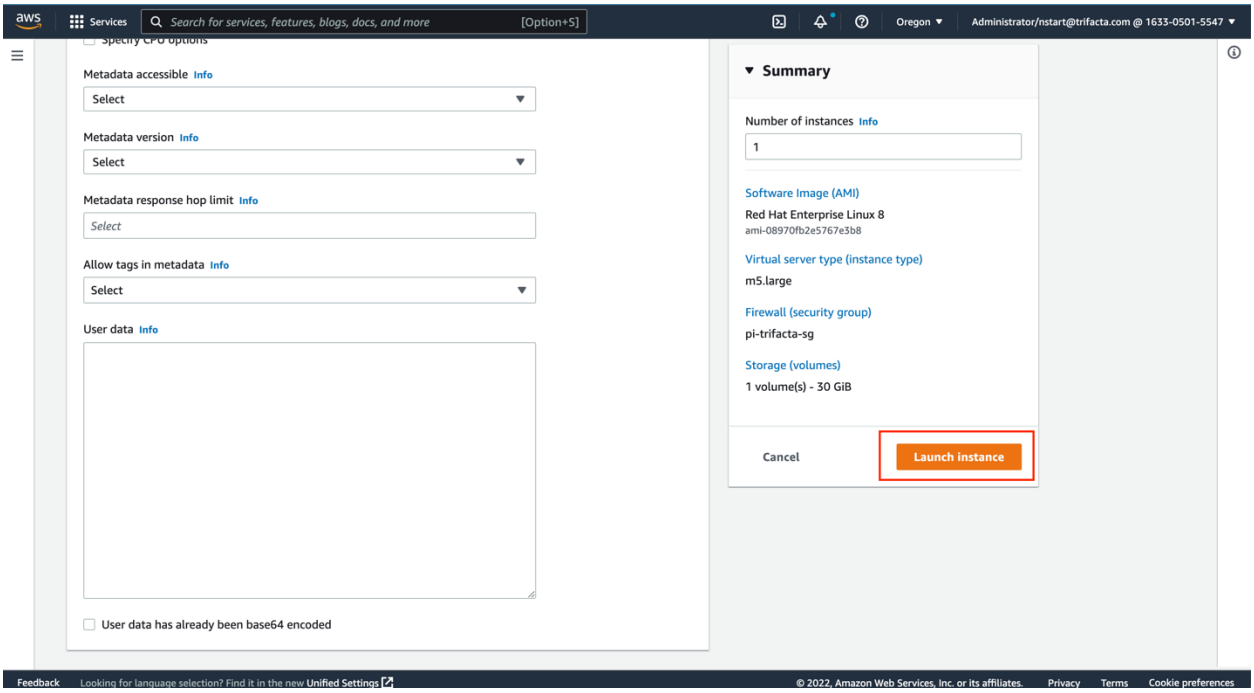
Create or use a policy: (see screenshot below for selected radio buttons and click "Next")



(Example Policy: note to replace items in red for your deployment)

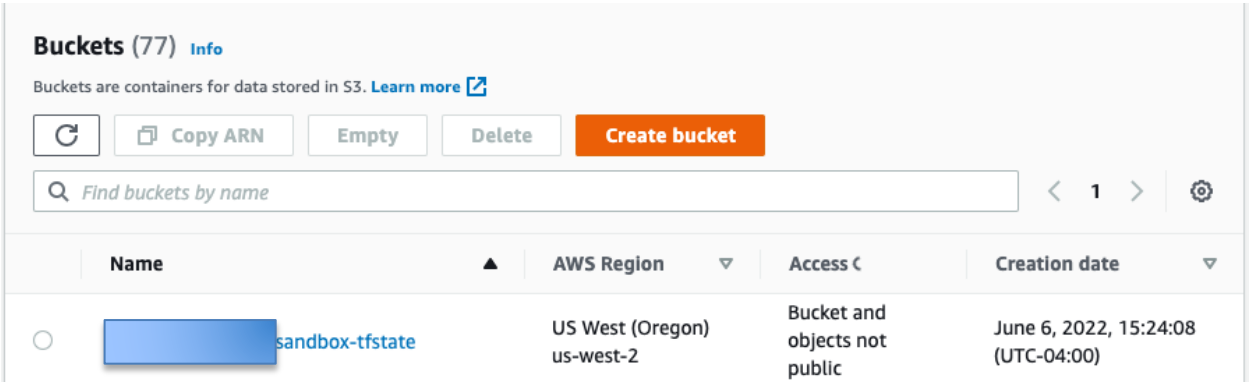
```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": [
            "s3:ListStorageLensConfigurations",
            "s3:ListAccessPointsForObjectLambda",
            "s3:GetAccessPoint",
            "s3:PutAccountPublicAccessBlock",
            "s3:GetAccountPublicAccessBlock",
            "s3:ListAccessPoints",
            "s3:PutAccessPointPublicAccessBlock",
            "s3:ListJobs",
            "s3:PutStorageLensConfiguration",
            "s3:ListMultiRegionAccessPoints",
            "s3:CreateJob",
            "s3:ListBucket"
         ],
         "Resource": "arn:aws:s3:::replace-with-your-bucket"
      },
      {
         "Sid": "VisualEditor1",
         "Effect": "Allow",
         "Action": "s3:*",
         "Resource": [
            "arn:aws:s3:::replace-with-your-bucket",
            "arn:aws:s3:::replace-with-your-bucket/*"
         ]
      }
   ]
}
```

10. Click Launch Instance

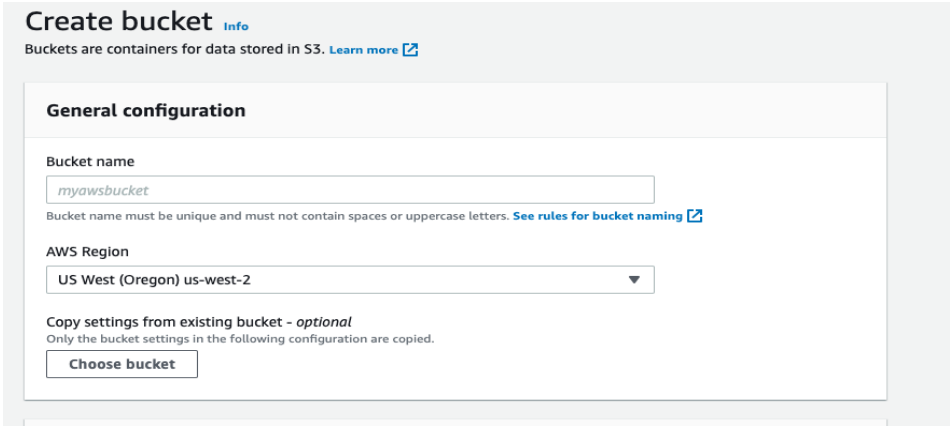


11. Create or use an existing S3 bucket:

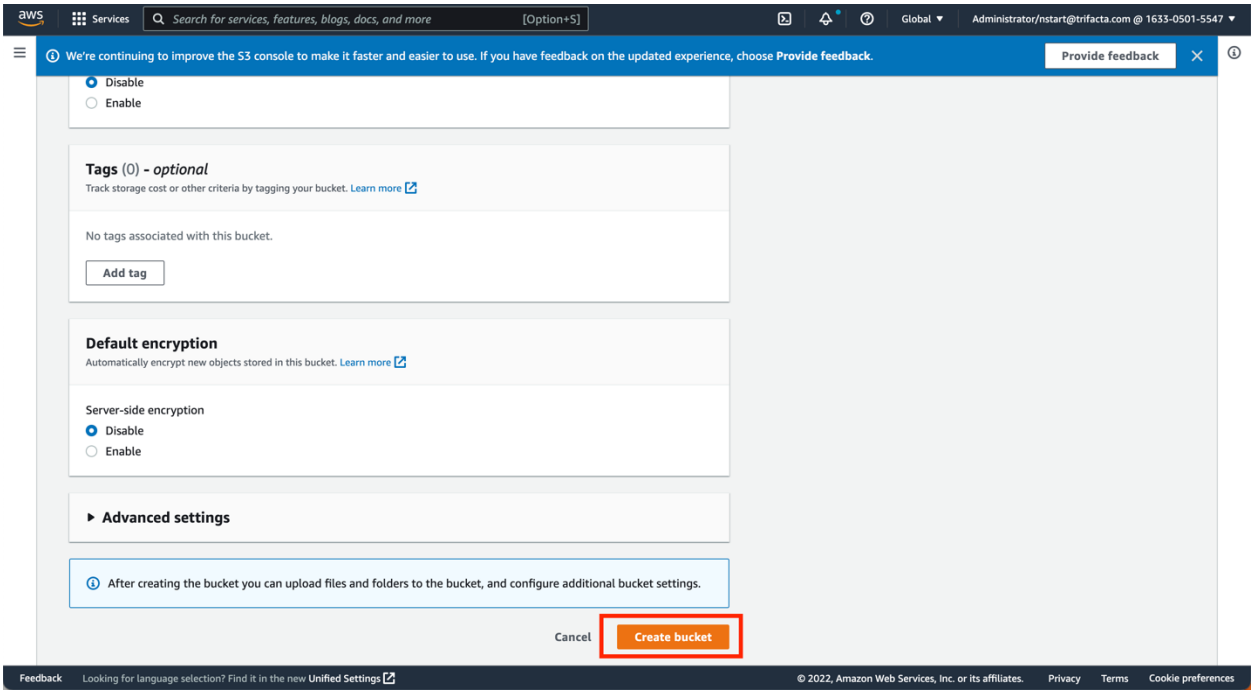
1. Create bucket



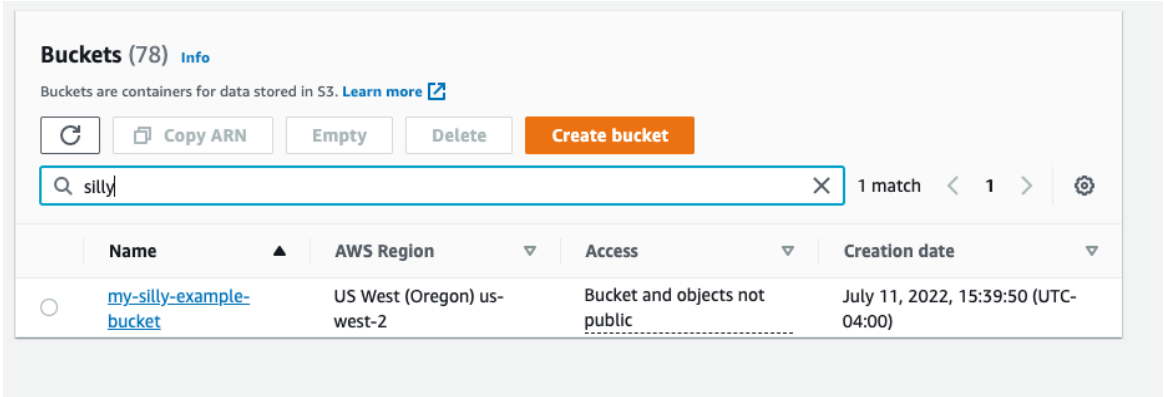
2. Give a name (globally unique) and select the same region as the deployed EC2 instance.



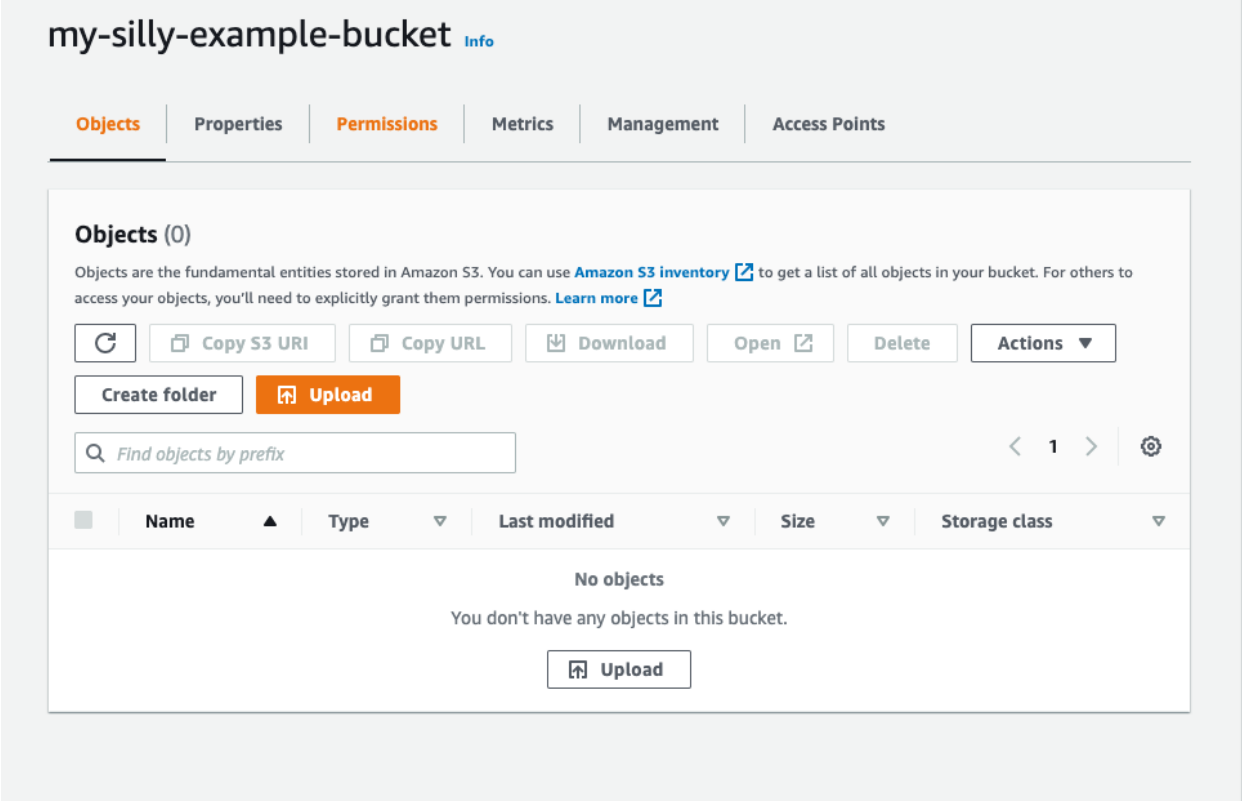
3. Click “Create Bucket”



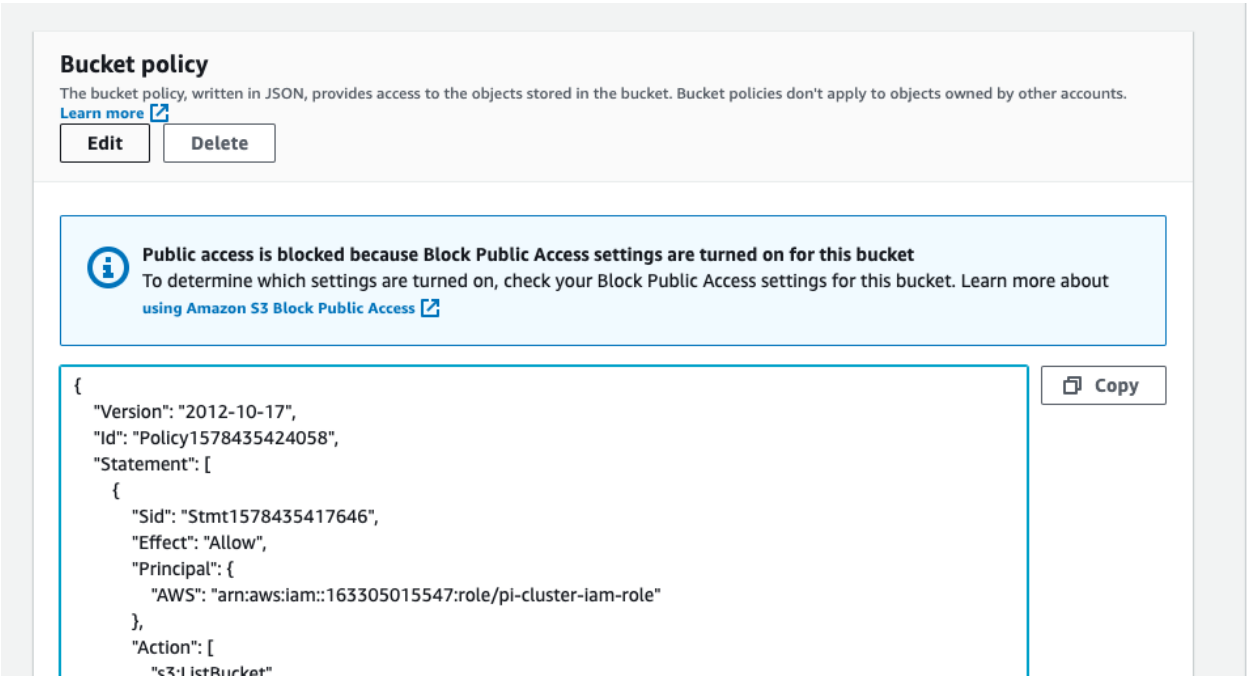
4. Then find the Bucket you created:



5. Click “Permissions”



6. Click “Edit” under Bucket Policy:



Example: note that the ARN is what we created earlier (the IAM Role for the Trifacta instance)
The bucket should be replaced with the bucket you created. You'll want to replace the red text with the arn number (**63305015547**), IAM Instance Profile (**cluster-iam-role**), and S3 bucket (**test-bucket**).

```
{
   "Version": "2012-10-17",
   "Id": "Policy1578435424058",
   "Statement": [
      {
         "Sid": "Stmt1578435417646",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::163305015547:role/ cluster-iam-role"
         },
         "Action": [
            "s3:ListBucket",
            "s3:GetBucketLocation"
         ],
         "Resource": "arn:aws:s3::: test-bucket"
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::163305015547:role/ cluster-iam-role"
         },
         "Action": "s3:*",
         "Resource": [
            "arn:aws:s3::: test-bucket",
            "arn:aws:s3::: test-bucket/*"
         ]
      }
   ]
}
```

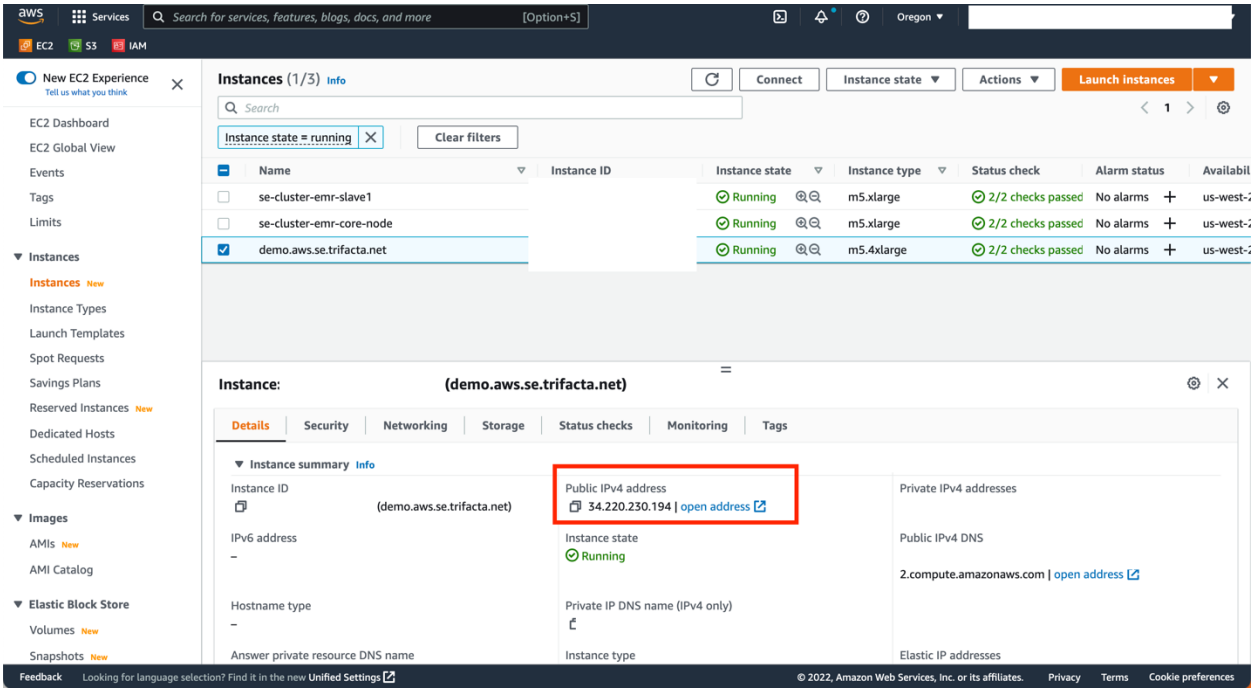
----------[CHECKPOINT]----------

So, at this point you have an EC2 instance, with an instance role, an S3 bucket and a policy that allows the role we created the ability to read/write to S3… but now we need to install Designer Cloud and configure it.

> Install Ansible, it will help make a vanilla install and configuration easier. If you don't have Python, PIP, or Ansible on your machine then click here to follow setup instructions for these.

What you will need:

1. Public IP address (from AWS Console EC2)



2. Pem key (created earlier) – remember where this was saved locally
3. Name of the S3 bucket Designer Cloud will be using
4. A license.json file (place the Designer Cloud license file in the ansible playbook directory - i.e. same location as the playbook.yml file location)

1. Edit the hosts file (change the IP and pem file location)

```
[trifacta]
trifacta_node  ansible_host=34.220.37.104

[trifacta:vars]
ansible_ssh_private_key_file=/Users/ppayandeh/Trifacta/keys/pi-keypair.pem
ansible_user=ec2-user
~
~
```

2. The configuration step:

```
.
├── config.txt
├── connect.sh
├── hosts
├── license.json
├── triconf_templates
│   ├── AWS_S3_only_config.template
│   └── Azure_ADLS_only_config.template
├── trifact_92_rhel7_playbook.yml
├── trifact_92_rhel8_playbook.yml
└── update_conf.py

1 directory, 9 files
```

copy the example/reference templates for the desired environment

cp ./triconf_templates/AWS_S3_only_config.template ./config.txt

The purpose of having a triconf_templates folder with a series of template files, is to create an inventory of different deployment scenarios. The goal is to minimize the scope of what needs to change in order to have trifacta-conf.json which is 1000+ lines of configuration and reduce it to: “what minimally needs to change for my deployment scenario to work?”. In our case with this document, we want to configure an AWS deployment with S3, so we copy the AWS_S3_only_config.template into config.txt (screenshot above). The file being named config.txt is just the convention and is expected from the Ansible playbook.

```
# IMPORTANT: Don't edit the file directly
#
# If you are doing an AWS deployment with S3 only, copy this file to config.txt
# cp AWS_S3_only_config.template config.txt
#
# Then change only the bucket you would like to configure Cloud Designer to use.
# If a line starts with #, it is a comment and will not alter the trifacta configuration.
# the JSON path is case sensitive so AWS.S3.BUCKET.NAME is not the same as aws.s3.bucket.name

# user config section

aws.s3.bucket.name = "bucket_name"

# END user config section


# SYSTEM CONFIG: do not edit
hdfs.enabled = False
webapp.storageProtocol = "s3"
aws.credentialProvider = "instance"
```

The only thing that needs to change for our simple AWS + S3 deployment is the S3 bucket name (aws.s3.bucket.name).

3. Place the license file in the directory where you unpacked the ansible configuration. It should look like this:

```
Trifacta/ansible/install_92 via 🐍 v3.8.7 on ☁️  (eu-west-1) on ☁️  ppayandeh@trifacta.com
❯ ls
config.txt                  hosts                       triconf_templates/          trifact_92_rhel8_playbook.yml
connect.sh*                 license.json                trifact_92_rhel7_playbook.yml  update_conf.py
```

4. Then run the playbook:
For Centos 8:
ansible-playbook -i hosts trifacta_92_rhel8_playbook.yml

Centos 7:
ansible-playbook -I hosts trifacta_92_rhel7_playbook.yml

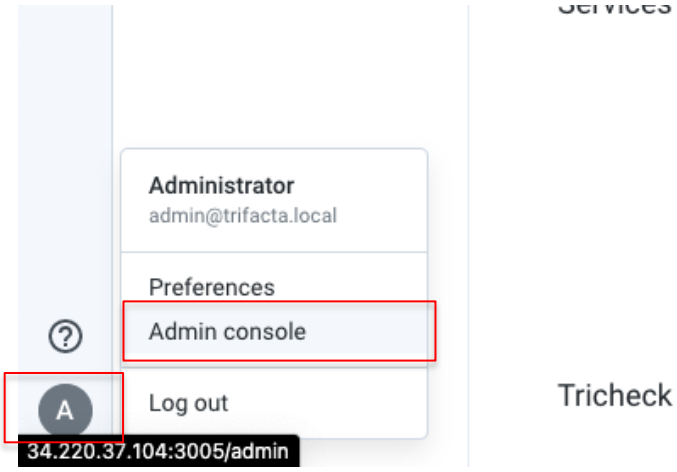
Which will output something like the following:

```
❯ ansible-playbook -i hosts playbook.yml

PLAY [all] *********************************************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************************************
The authenticity of host '34.220.37.104 (34.220.37.104)' can't be established.
ECDSA key fingerprint is SHA256:ZJLbnTNMA0b+cN5NqwDWRuyklMmd1ul1VreyskGUL6o.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
ok: [trifacta_node]

TASK [Create /root/trifacta_temp if it does not exist] *************************************************************************************************************************
changed: [trifacta_node]

TASK [Download dependencies file (trifacta-server-deps-9.2.0-centos8.tar.gz)] **************************************************************************************************
changed: [trifacta_node]

TASK [Download trifacta RPM (trifacta-server-9.2.0-184.centos8.x86_64.rpm)] ****************************************************************************************************
changed: [trifacta_node]

TASK [Unarchive (trifacta-server-deps-9.2.0-centos8.tar.gz)] *******************************************************************************************************************
changed: [trifacta_node]

TASK [copy local.repo] *********************************************************************************************************************************************************
changed: [trifacta_node]

TASK [trifacta repo directory status] ******************************************************************************************************************************************
ok: [trifacta_node]

TASK [trifacta repo directory status] ******************************************************************************************************************************************
ok: [trifacta_node]

TASK [Move trifacta repo directory] ********************************************************************************************************************************************
changed: [trifacta_node]

TASK [install python3] *********************************************************************************************************************************************************
changed: [trifacta_node]

TASK [install java 8 from local repo] ******************************************************************************************************************************************
changed: [trifacta_node]

TASK [postgress dnf install] ***************************************************************************************************************************************************
changed: [trifacta_node]

TASK [postgress client dnf install] ********************************************************************************************************************************************
ok: [trifacta_node]

TASK [install nodejs, nginx] ***************************************************************************************************************************************************
changed: [trifacta_node]

TASK [install trifacta] ********************************************************************************************************************************************************
changed: [trifacta_node]

TASK [copy license file] *******************************************************************************************************************************************************
changed: [trifacta_node]

TASK [encrypt key] *************************************************************************************************************************************************************
changed: [trifacta_node]

TASK [Check if PostgreSQL database is initialized.] ****************************************************************************************************************************
ok: [trifacta_node]

TASK [init db] *****************************************************************************************************************************************************************
changed: [trifacta_node]

TASK [backup /var/lib/pgsql/12/data/pg_hba.conf  /var/lib/pgsql/12/data/pg_hba.conf.bak] ***************************************************************************************
changed: [trifacta_node]

TASK [cp /opt/trifacta/bin/setup-utils/db/pg_hba.conf.SAMPLE /var/lib/pgsql/12/data/pg_hba.conf] ******************************************************************************
changed: [trifacta_node]

TASK [Start service postgresql-12, if not running] *****************************************************************************************************************************
changed: [trifacta_node]

TASK [initialize the database] *************************************************************************************************************************************************
changed: [trifacta_node]

TASK [Start service postgresql-12, if not running] *****************************************************************************************************************************
skipping: [trifacta_node]

TASK [change permisions to fix issue [JIRA]] ***********************************************************************************************************************************
changed: [trifacta_node]

TASK [copy update_conf.py] *****************************************************************************************************************************************************
changed: [trifacta_node]

TASK [run update conf] *********************************************************************************************************************************************************
changed: [trifacta_node]

TASK [Start trifacta service, if not running] **********************************************************************************************************************************
changed: [trifacta_node]

PLAY RECAP *********************************************************************************************************************************************************************
trifacta_node              : ok=26   changed=22   unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

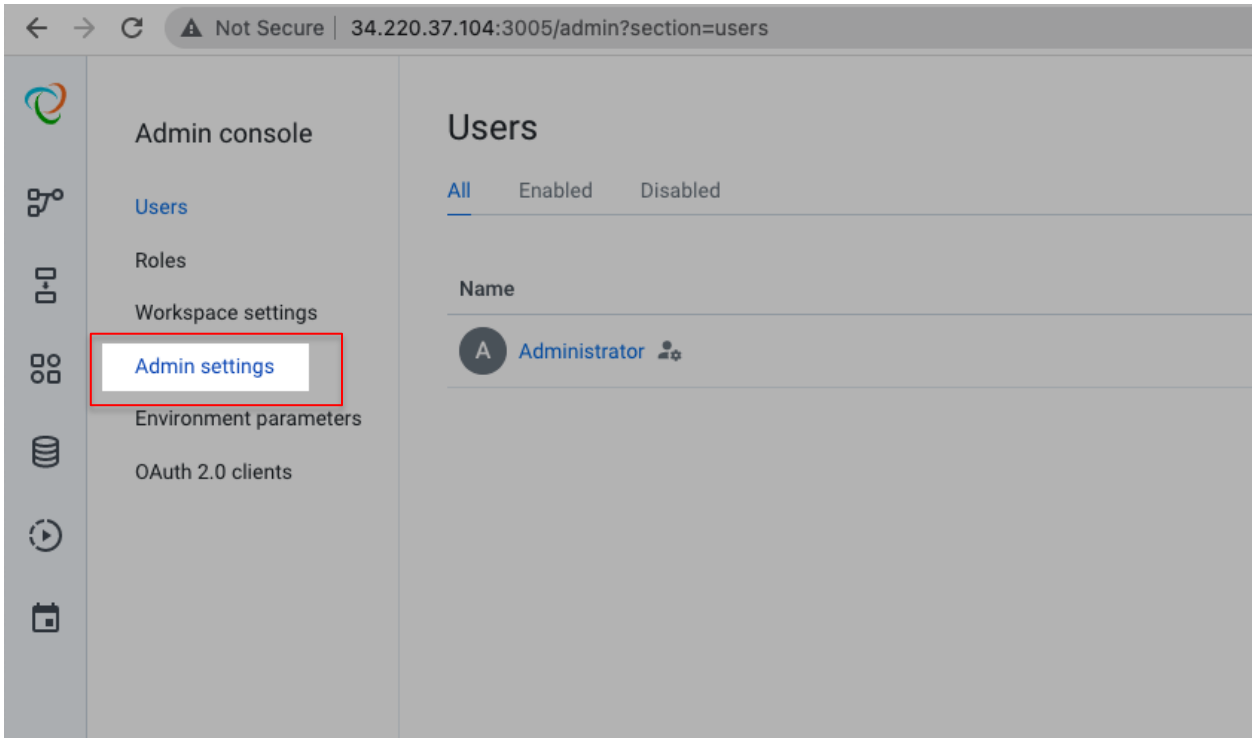
Now you need to open a browser and point it at: **http:<designer-cloud ip address>:3005**
NOTE: The Designer Cloud IP Address is the Public IP address from AWS Console EC2.

** The default username and password is:
Username: admin@trifacta.local
password : admin

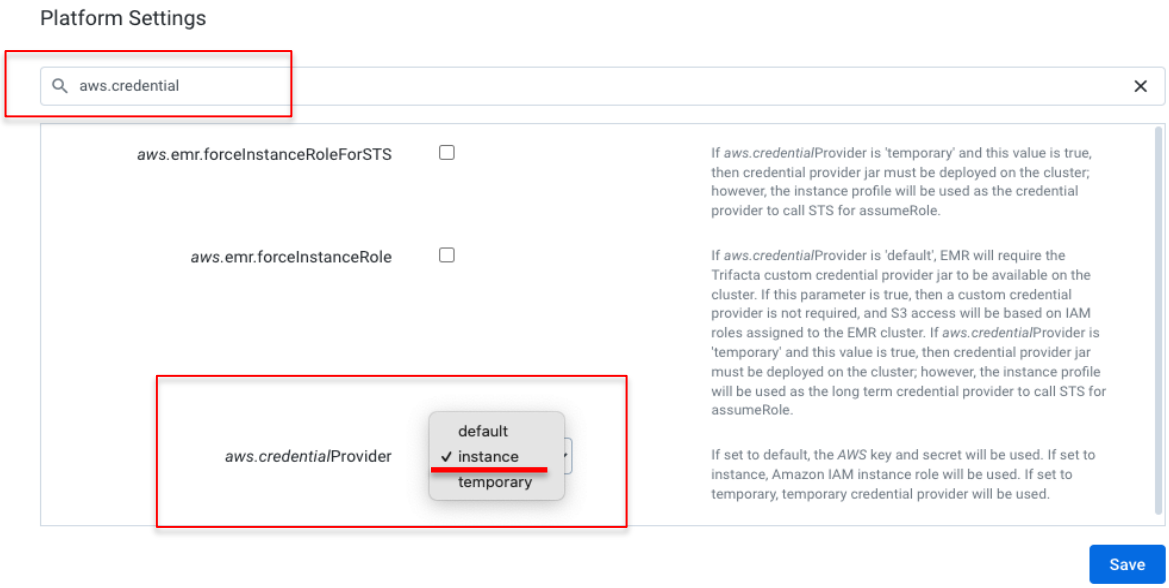
1. In the Designer Cloud UI, click the “A” on the very bottom left side of the UI -> then click “Admin console”



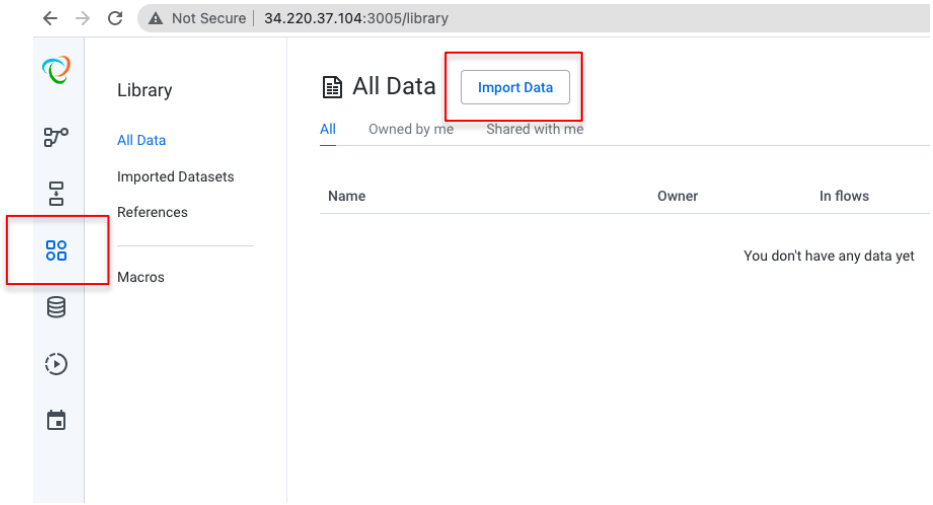
2. Then click “Admin settings”



3. In the search bar under Platform Settings, search for: aws.credential (to filter) Change The aws.credentialProvider from “default” to “instance” and click “Save”. This will take a few minutes to restart the Designer Cloud service. When the service comes back online, the issue should be fixed and you should have access to S3.



4. If you go to Library (little icon on the far left side of the UI) and click on “Import Data”



5. You should see your S3 bucket like the following: