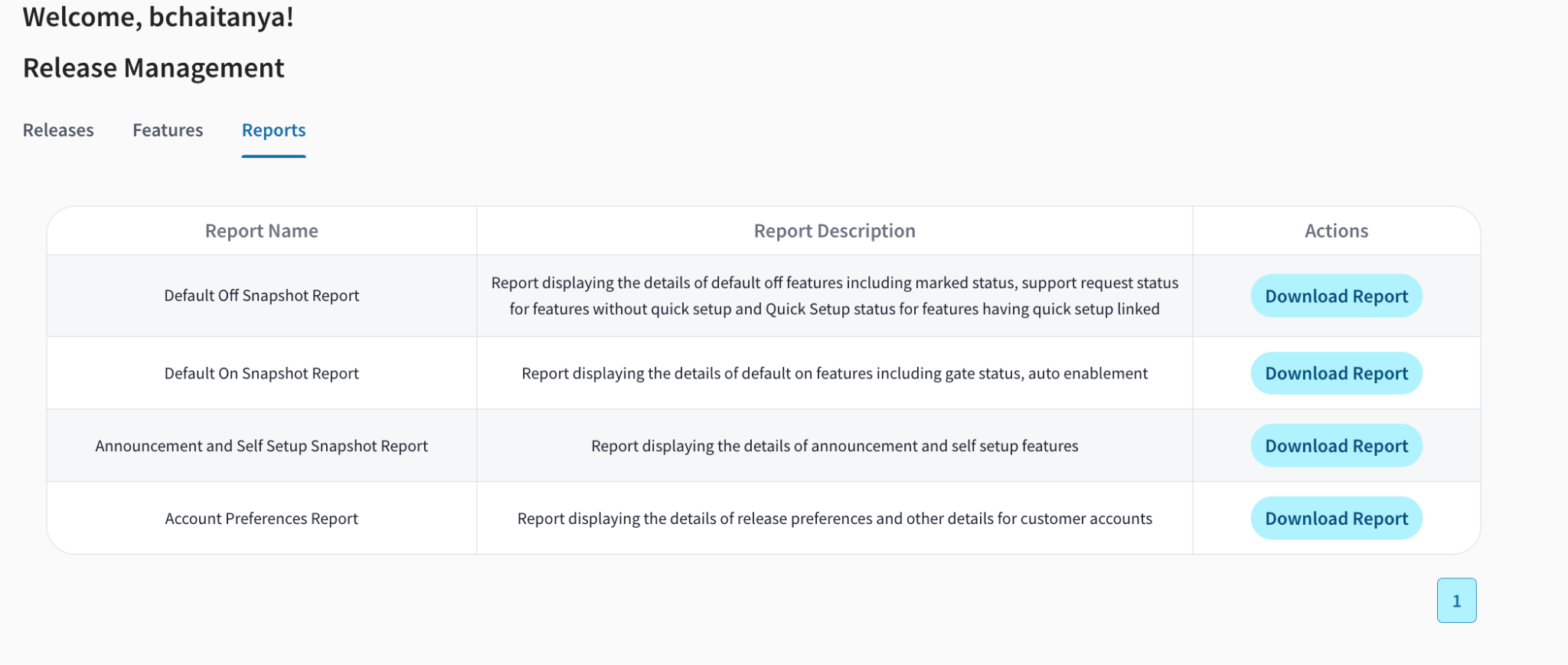
1. Get IDs from <https://stage.eightfold.ai/internal/console/release_management?tab_id=release_management_reports>

PS: use account preferences report

1. Raise request on #customer-account-access
2. Run script to add permissions

python modify\_perms\_in\_config.py --group\_regex costar-sandbox.com --roles admin --permissions perm\_manage\_rectruitmentanalytics\_dashboard --add --user bchaitanya@eightfold.ai



1. Run Script to create dataset

from quicksight import qs\_data

from config import config

from utils.os\_constants import EF\_DEFAULT\_REGION

group\_ids = ['10xgenomics.com']

error\_group\_ids = [] # Create an empty list to store erroring group IDs

for group\_id in group\_ids:

db\_shard\_config = config.get\_config\_for\_partition('db\_group\_shard\_config', group\_id)

try:

account\_region = list(db\_shard\_config.keys())[0]

except:

continue

if account\_region != EF\_DEFAULT\_REGION:

print(f"Skipping group\_id {group\_id} as account\_region {account\_region} is different from {EF\_DEFAULT\_REGION}")

continue

try:

handler = qs\_data.QuicksightDataHandler(group\_id)

registry = handler.get\_dataset\_provisioning\_registry()

registry["ta\_feature\_campaign\_application\_activity"]["create"]()

except Exception as e: # Catch any exceptions (more specific than just Exception)

print(f"Error creating dataset setup for group ID '{group\_id}': {e}")

error\_group\_ids.append(group\_id)



1. Run script update dashboard

from quicksight import qs\_dashboards

from config import config

from utils.os\_constants import EF\_DEFAULT\_REGION

dashboard\_key = 'tafeatureusage'

group\_ids = ['10xgenomics.com']

error\_group\_ids = [] # Create an empty list to store erroring group IDs

for group\_id in group\_ids:

db\_shard\_config = config.get\_config\_for\_partition('db\_group\_shard\_config', group\_id)

try:

account\_region = list(db\_shard\_config.keys())[0]

except:

continue

if account\_region != EF\_DEFAULT\_REGION:

print(f"Skipping group\_id {group\_id} as account\_region {account\_region} is different from {EF\_DEFAULT\_REGION}")

continue

try:

handler = qs\_dashboards.QuicksightDashboardHandler(group\_id)

handler.create\_dashboard\_setup(dashboard\_key)

except Exception as e: # Catch any exceptions (more specific than just Exception)

print(f"Error creating dashboard setup for group ID '{group\_id}': {e}")

error\_group\_ids.append(group\_id)

