ONBOARDING

User and Groups API START GUIDE

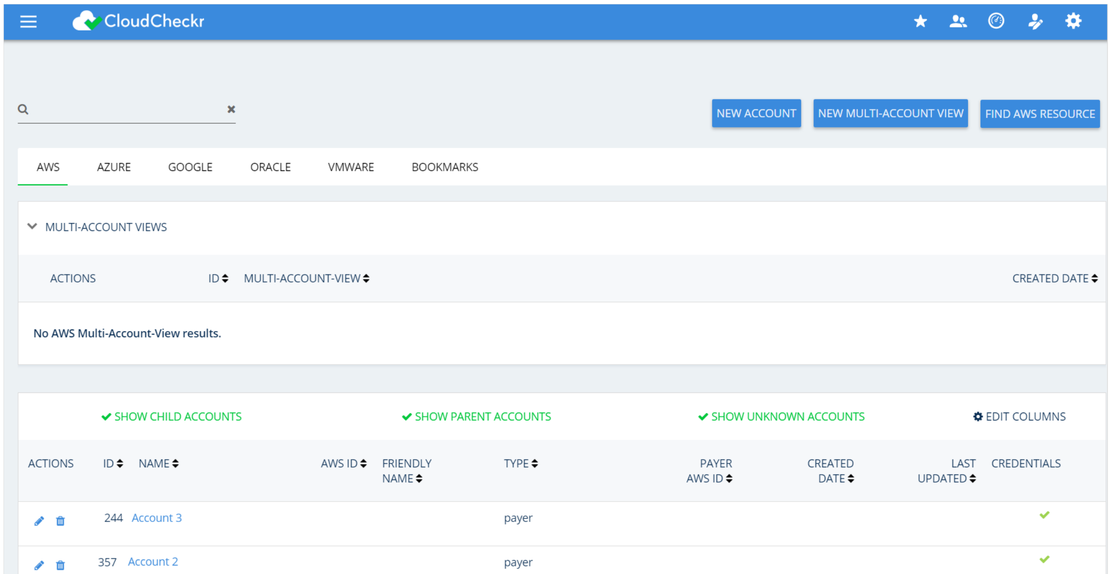
CloudCheckr

Create a Group Permission Template

The first Group will be created through the UI in order to set up the permission template. This group's permissions will then be used as a template for future groups added through the API.

1. Log into CloudCheckr with the Administrator role.

   The Accounts List page opens.

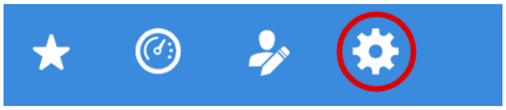


ACTIONS	ID	MULTI-ACCOUNT-VIEW	CREATED DATE

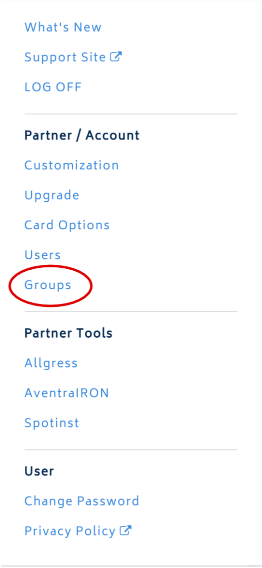
No AWS Multi-Account-View results.

ACTIONS	ID	NAME	AWS ID	FRIENDLY NAME	TYPE	PAYER AWS ID	CREATED DATE	LAST UPDATED	CREDENTIALS
	244	Account 3			payer				✓
	357	Account 2			payer				✓

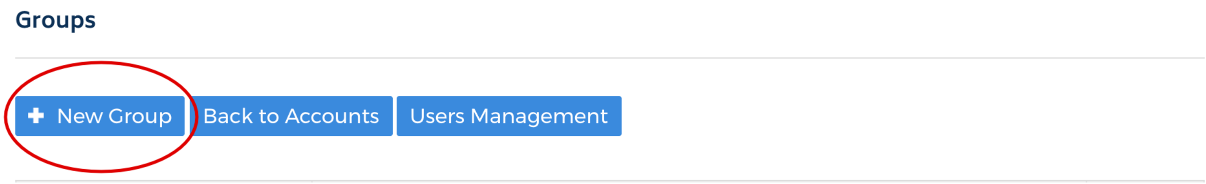
2. From the menu bar, click the **Settings** icon.



The **Settings** drop down expands and select Groups.

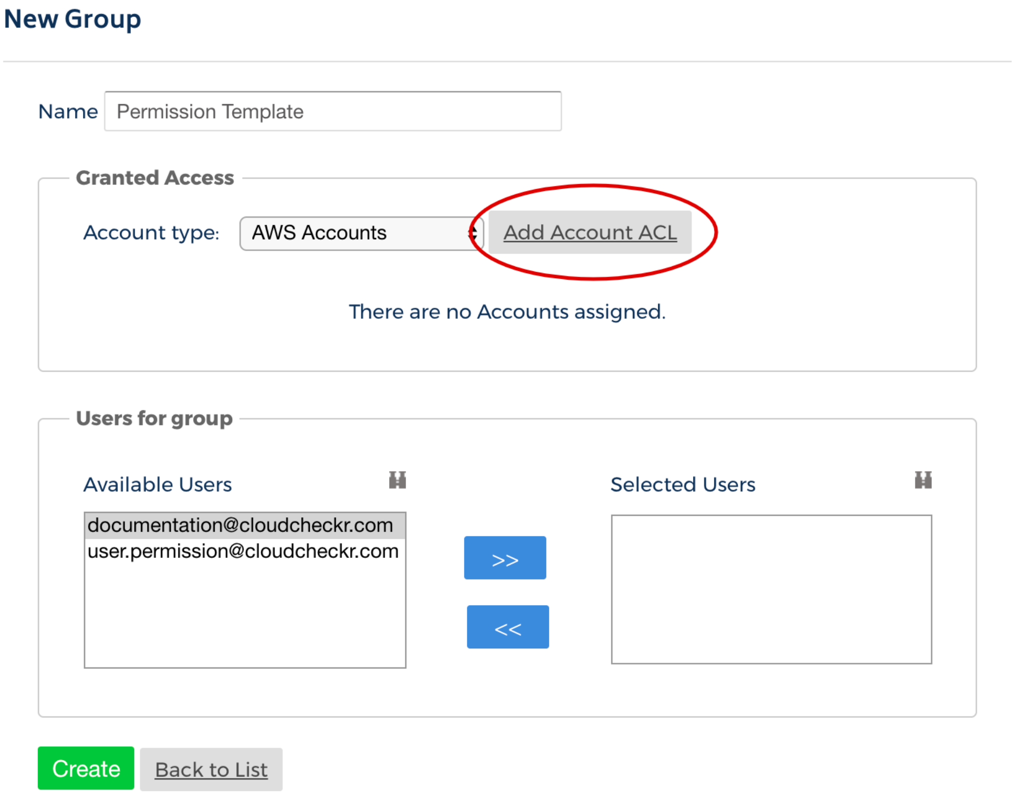


3. Add a new group. This group will act as a template.



Groups

4. Name the Group **Permission Template** and write down this name. Add an initial ACL for an AWS Account.



New Group

Name: Permission Template

Granted Access

Account type: AWS Accounts

There are no Accounts assigned.

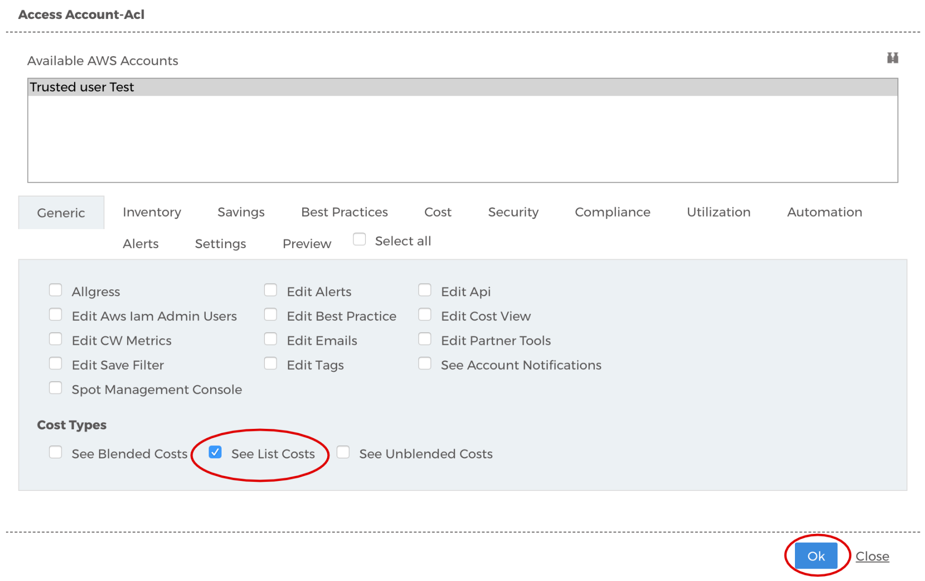
Users for group

Available Users

documentation@cloudcheckr.com
user.permission@cloudcheckr.com

Selected Users

5. Select an AWS Account as an example. Select the permissions that you would like to enable. For example, you can select See List Cost and leave Unblended and Blended cost deselected, so the users in the group will only see List Cost. You can go through all of tabs to identify all of the permissions that you would like to enable.



Access Account-Acl

Available AWS Accounts

Trusted user Test

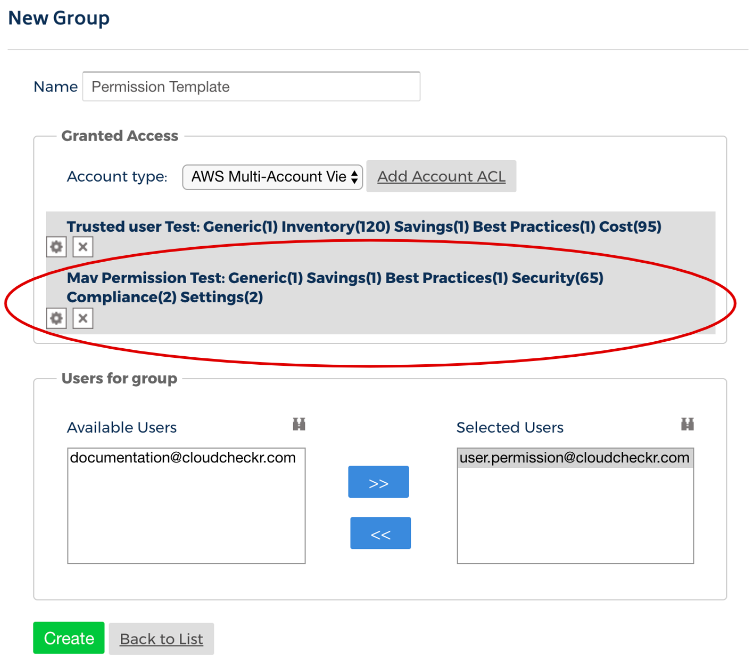
□ Select all

□ Allgress
□ Edit Aws Iam Admin Users
□ Edit CW Metrics
□ Edit Save Filter
□ Spot Management Console
□ Edit Alerts
□ Edit Best Practice
□ Edit Emails
□ Edit Tags
□ Edit Api
□ Edit Cost View
□ Edit Partner Tools
□ See Account Notifications

Cost Types

□ See Blended Costs ☑ See List Costs □ See Unblended Costs

6. Select **Ok** to add the permissions for this group.
7. *Optional*: Repeat step 5 for the three other kinds of Accounts in CloudCheckr (AWS MAVs, Azure Accounts, and Azure MAVs). The ACLs are unique to all four types of accounts. The example below has an additional ACL for an AWS MAV. The name of the accounts or MAVs should be written down for a future step.



New Group

Name: Permission Template

Granted Access

Account type: AWS Multi-Account Vie

Trusted user Test: Generic(1) Inventory(120) Savings(1) Best Practices(1) Cost(95)

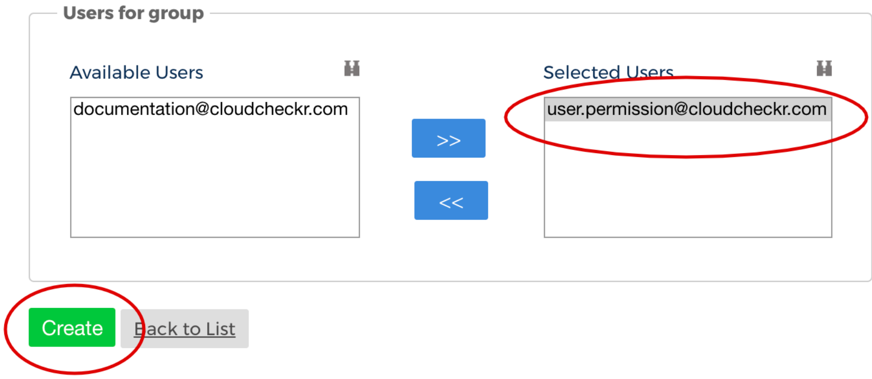
Mav Permission Test: Generic(1) Savings(1) Best Practices(1) Security(65) Compliance(2) Settings(2)

Users for group

Available Users: documentation@cloudcheckr.com

Selected Users: user.permission@cloudcheckr.com

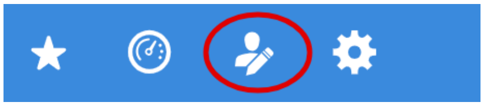
8. *Optional*: Add a user to the group and login as that user to verify that the permissions are what was intended.
9. Select **Create** to create the group.



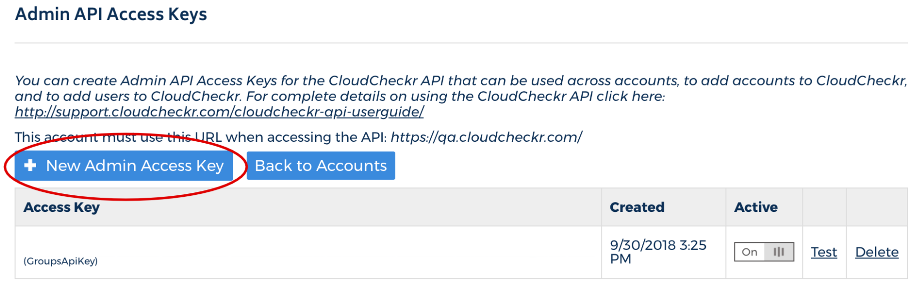
Get Group Permissions (ACLs) With The API

The later groups will be created with the API.

1. Navigate to Admin Functions and select Admin API Key.



2. Create Admin PI Key.



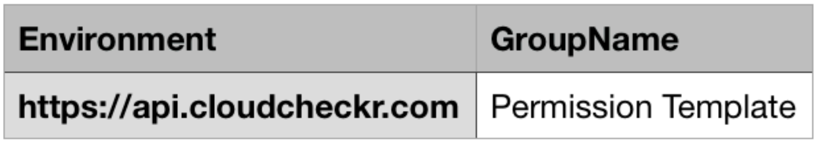
Admin API Access Keys

You can create Admin API Access Keys for the CloudCheckr API that can be used across accounts, to add accounts to CloudCheckr, and to add users to CloudCheckr. For complete details on using the CloudCheckr API click here: http://support.cloudcheckr.com/cloudcheckr-api-userguide/

This account must use this URL when accessing the API: *https://qa.cloudcheckr.com/*

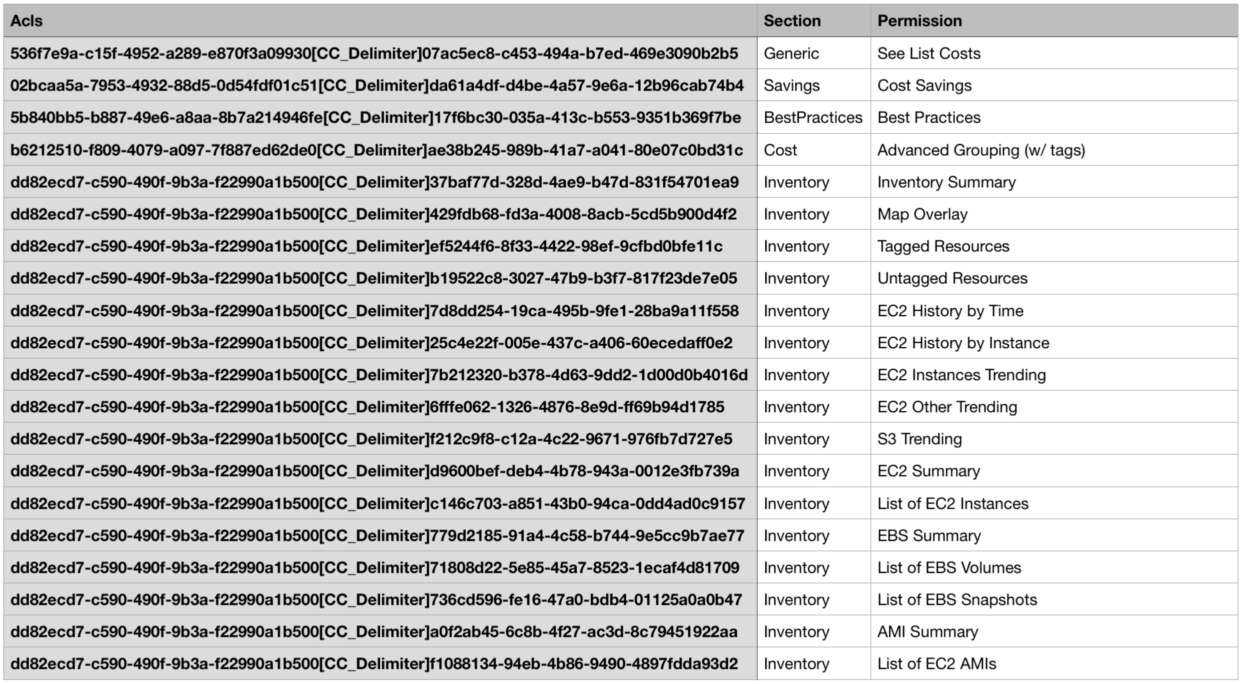
Access Key	Created	Active		
(GroupsApiKey)	9/30/2018 3:25 PM	On	Test	Delete

3. Navigate to the GetAcls Folder. Open up the permission_template_input.csv file.
4. Enter in the environment in the first column. Defaults to https://api.cloudcheckr.com, but this can vary depending on if you are in the eu, au, or Self-Hosted version of CloudCheckr. (If you are using a Self-Hosted version of CloudCheckr you should add the url to the check_invalid_env function for proper error checking)
5. Enter in the Group Name in the second column of the Permission Template Group. In the above example this was **Permission Template**.



Environment	GroupName
https://api.cloudcheckr.com	Permission Template

6. Run the python program to get the Acls in a csv file.
7. Run “python get_access_control_list_per_group.py 0000000000000”
8. The admin api access key should be entered via the command line after the file. The 0000000 in the example above should be replaced with an Admin API Access Key.
9. This will download the permissions for each of the 4 account types that are available in the group. (AWS Accounts, AWS MAV Accounts, Azure Accounts, Azure MAV Accounts).
10. If there is more than one account type in the group template, the program will use the permissions in the FIRST group and skip the remaining ones.
11. This will output one csv file for every account type that there was an input for. The first file will be AwsAccountPermissions.csv and will be a list of permissions that correspond to the template.

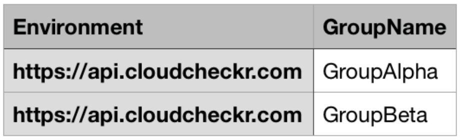


Acls	Section	Permission
536f7e9a-c15f-4952-a289-e870f3a09930[CC_Delimiter]07ac5ec8-c453-494a-b7ed-469e3090b2b5	Generic	See List Costs
02bcaa5a-7953-4932-88d5-0d54fdf01c51[CC_Delimiter]da61a4df-d4be-4a57-9e6a-12b96cab74b4	Savings	Cost Savings
5b840bb5-b887-49e6-a8aa-8b7a214946fe[CC_Delimiter]17f6bc30-035a-413c-b553-9351b369f7be	BestPractices	Best Practices
b6212510-f809-4079-a097-7f887ed62de0[CC_Delimiter]ae38b245-989b-41a7-a041-80e07c0bd31c	Cost	Advanced Grouping (w/ tags)
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]37baf77d-328d-4ae9-b47d-831f54701ea9	Inventory	Inventory Summary
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]429fdb68-fd3a-4008-8acb-5cd5b900d4f2	Inventory	Map Overlay
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]ef5244f6-8f33-4422-98ef-9cfbd0bfe11c	Inventory	Tagged Resources
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]b19522c8-3027-47b9-b3f7-817f23de7e05	Inventory	Untagged Resources
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]7d8dd254-19ca-495b-9fe1-28ba9a11f558	Inventory	EC2 History by Time
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]25c4e22f-005e-437c-a406-60ecedaff0e2	Inventory	EC2 History by Instance
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]7b212320-b378-4d63-9dd2-1d00d0b4016d	Inventory	EC2 Instances Trending
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]6fffe062-1326-4876-8e9d-ff69b94d1785	Inventory	EC2 Other Trending
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]f212c9f8-c12a-4c22-9671-976fb7d727e5	Inventory	S3 Trending
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]d9600bef-deb4-4b78-943a-0012e3fb739a	Inventory	EC2 Summary
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]c146c703-a851-43b0-94ca-0dd4ad0c9157	Inventory	List of EC2 Instances
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]779d2185-91a4-4c58-b744-9e5cc9b7ae77	Inventory	EBS Summary
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]71808d22-5e85-45a7-8523-1ecaf4d81709	Inventory	List of EBS Volumes
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]736cd596-fe16-47a0-bdb4-01125a0a0b47	Inventory	List of EBS Snapshots
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]a0f2ab45-6c8b-4f27-ac3d-8c79451922aa	Inventory	AMI Summary
dd82ecd7-c590-490f-9b3a-f22990a1b500[CC_Delimiter]f1088134-94eb-4b86-9490-4897fdda93d2	Inventory	List of EC2 AMIs

12. The number of files corresponds to the number of account types in the template with a maximum of 4. (AwsPermissions, AwsMavPermissions, AzurePermissions, AzureMavPermissions).
13. These files should **copied** to the folder AddGroupPermissions.
14. You can open each of them up to see the section and permission that the acl corresponds to.

Create Groups with the API

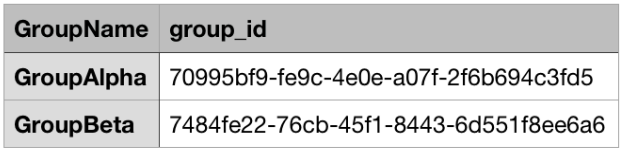
1. Navigate to the folder CreateGroups. Open up the create_groups_input.csv file.
2. Enter in the environment in the first column. Defaults to https://api.cloudcheckr.com, but this can vary depending on if you are in the eu, au, or Self-Hosted version of CloudCheckr. This should be the same for all entries of the first column.
3. Enter in the name of desired groups in the second column.
4. These group names *must* be unique. If there are any duplicate group names, the program will end and recommend that you rename group names to make them unique. Historically this was allowed by CloudCheckr, but is no longer allowed going forward.



Environment	GroupName
https://api.cloudcheckr.com	GroupAlpha
https://api.cloudcheckr.com	GroupBeta

Enter in group names in second column

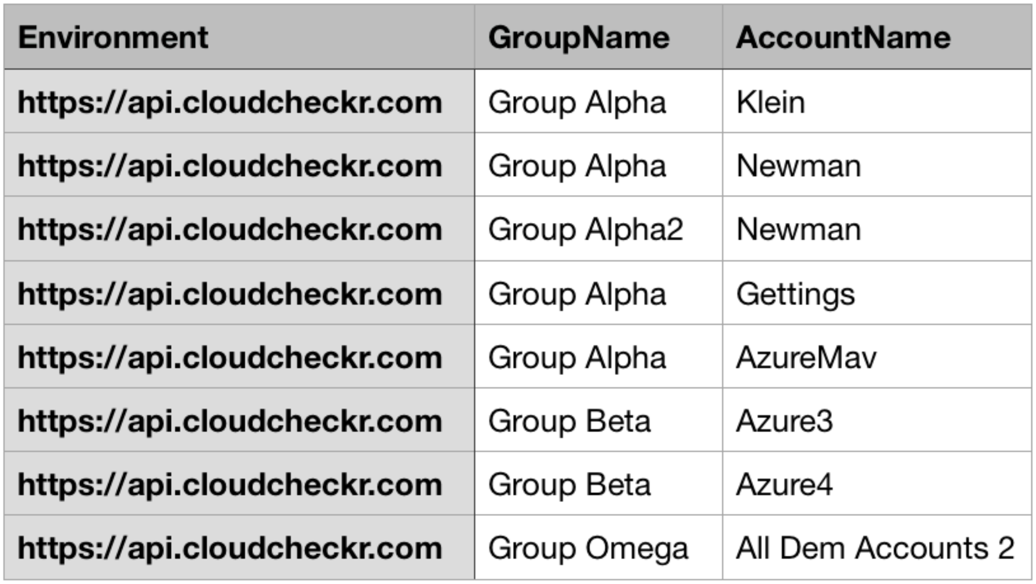
5. Run the python program create_groups.py
6. Run "python create_groups.py 00000000" Where the Admin API Key goes after the name of the python file. It uses the create_group Admin API call.
7. This will output the GroupsCreated.csv file. The first column is the name of the group. The second column is the group id.



GroupName	group_id
GroupAlpha	70995bf9-fe9c-4e0e-a07f-2f6b694c3fd5
GroupBeta	7484fe22-76cb-45f1-8443-6d551f8ee6a6

Add Permissions To Groups

1. Navigate to the AddGroupPermissions folder.
2. Make sure that you **copied** over the csv files that were previously generated (maxiumum of 4). Also remember to **delete** any files that were previously there, but there is no longer an account type for.
3. Open up the group_permissions_input.csv file.

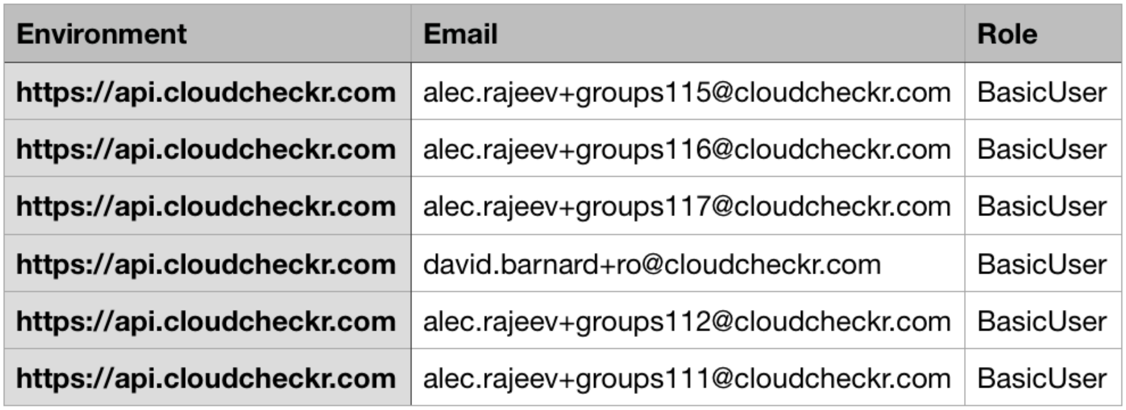


Environment	GroupName	AccountName
https://api.cloudcheckr.com	Group Alpha	Klein
https://api.cloudcheckr.com	Group Alpha	Newman
https://api.cloudcheckr.com	Group Alpha2	Newman
https://api.cloudcheckr.com	Group Alpha	Gettings
https://api.cloudcheckr.com	Group Alpha	AzureMav
https://api.cloudcheckr.com	Group Beta	Azure3
https://api.cloudcheckr.com	Group Beta	Azure4
https://api.cloudcheckr.com	Group Omega	All Dem Accounts 2

4. Enter the environment in the first column.
5. Enter the name of the group you would like to add a permission to to the second column. (Note: This name is used to identify the group in the python script.)
6. In the third column, add the name of the Account in CloudCheckr.
7. Then run the python script to add the acls.
8. Run python3 add_group_permissions.py 00000000 where the 0000 corresponds to the Admin Api Key.

Add Users

1. Navigate to the AddUsers folder. Open up the add_users_input.csv file.
2. Enter the environment in the first column.
3. Enter the user's email in the second column.
4. Enter the user's role in the third column. Available options are ReadOnlyUser, BasicUser, BasicPlusUser, User, Administrator.

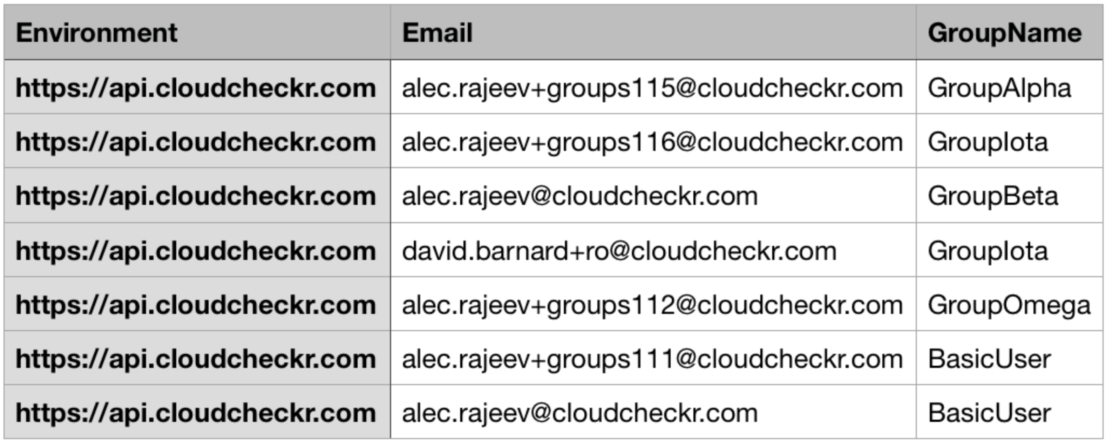


Environment	Email	Role
https://api.cloudcheckr.com	alec.rajeev+groups115@cloudcheckr.com	BasicUser
https://api.cloudcheckr.com	alec.rajeev+groups116@cloudcheckr.com	BasicUser
https://api.cloudcheckr.com	alec.rajeev+groups117@cloudcheckr.com	BasicUser
https://api.cloudcheckr.com	david.barnard+ro@cloudcheckr.com	BasicUser
https://api.cloudcheckr.com	alec.rajeev+groups112@cloudcheckr.com	BasicUser
https://api.cloudcheckr.com	alec.rajeev+groups111@cloudcheckr.com	BasicUser

5. Run the python script to add users.
6. Run python add_users.py 000000 where the 0000000 corresponds to the Admin API Key.

Add Users to Groups

1. Navigate to the AddUsersToGroups folder. Open up the add_users_to_groups.csv file.
2. Add the environment in the first column.
3. Add the email in the second column. This email must be pre-created in this account.
4. Add the group name in the third column.



Environment	Email	GroupName
https://api.cloudcheckr.com	alec.rajeev+groups115@cloudcheckr.com	GroupAlpha
https://api.cloudcheckr.com	alec.rajeev+groups116@cloudcheckr.com	GroupIota
https://api.cloudcheckr.com	alec.rajeev@cloudcheckr.com	GroupBeta
https://api.cloudcheckr.com	david.barnard+ro@cloudcheckr.com	GroupIota
https://api.cloudcheckr.com	alec.rajeev+groups112@cloudcheckr.com	GroupOmega
https://api.cloudcheckr.com	alec.rajeev+groups111@cloudcheckr.com	BasicUser
https://api.cloudcheckr.com	alec.rajeev@cloudcheckr.com	BasicUser

5. Remove all duplicate group names. The script will not run until duplicate group names are removed. They are no longer allowed.
6. Run the python script to add users to groups.
7. Run python add_users_to_groups.py 00000 where the 0000 corresponds to the Admin API Key.

Learn more about the CloudCheckr
Cloud Management Platform at
www.cloudcheckr.com.