



# CTX-AWS User Guide

## Contents

---

CTX-AWS User Guide.....	1
Contents.....	2
Versions .....	4
Document Revisions.....	4
Module Versions .....	4
Preface .....	5
About this Manual .....	5
Audience .....	5
Related Material.....	5
Abbreviations used in this Document .....	5
Requirements.....	6
Integration .....	7
Integration with Third-Party Systems.....	7
Creating IAM Users .....	7
Creating IAM Groups and Attaching Policies .....	9
Adding Users to Groups.....	10
Integrating with Existing Infrastructure .....	11
1    AWS-EC2-CREATE-INSTANCE .....	12
1.1    Overview .....	12
1.2    Inputs .....	12
1.3    Outputs.....	12
2    AWS-EC2-GET-INSTANCE .....	13
2.1    Overview .....	13
2.2    Inputs .....	13
2.3    Outputs.....	13
3    AWS-EC2-START-INSTANCE.....	16
3.1    Overview .....	16
This subtask starts an instance in a region. ....	16
3.2    Inputs .....	16
3.3    Outputs.....	16
4    AWS-EC2-STOP-INSTANCE .....	17
4.1    Overview .....	17
This subtask stops an instance in a region. ....	17
4.2    Inputs .....	17
4.3    Outputs.....	17
5    AWS-EC2-RESTART-INSTANCE.....	18
5.1    Overview .....	18
This subtask restarts an instance in a region. ....	18
5.2    Inputs .....	18
5.3    Outputs.....	18
6    AWS-EC2-TERMINATE-INSTANCE.....	19
6.1    Overview .....	19
6.2    Inputs .....	19

6.3	Outputs .....	19
7	AWS-EC2-CHANGE-INSTANCE-TYPE .....	20
7.1	Overview .....	20
7.2	Inputs .....	20
7.3	Outputs .....	20
8	AWS-EC2-CREATE-CPU-ALARM-FOR-INSTANCE .....	21
8.1	Overview .....	21
8.2	Inputs .....	21
8.3	Check if CPU alarm is created.....	21
9	AWS-CE-GET-COST .....	22
9.1	Overview .....	22
	Retrieves cost and usage metrics for your account. ....	22
9.2	Inputs .....	22
9.3	Outputs.....	22
10	AWS-IAM-GET-USERS .....	23
10.1	Overview .....	23
	Get all users on AWS account and their roles and policies. ....	23
10.2	Inputs .....	23
10.3	Outputs.....	23

## Versions

---

### Document Revisions

The following revisions have been made to this document

Date	Revision	Notes
14/03/2018	0.1	First Draft
24/09/2018	1.0	Updates to document to include additional functionality

### Module Versions

The following revisions have been made to this document

Date	Revision	Notes
14/03/2018	1.0	Creation of: <ul style="list-style-type: none"><li>• Create Instance</li><li>• Get Instance</li><li>• Start Instance</li><li>• Stop Instance</li><li>• Restart Instance</li><li>• Terminate Instance</li><li>• Change Instance</li><li>• Create CPU Alarm for Instance</li></ul>
25/09/2018	1.1	Creation of: <ul style="list-style-type: none"><li>• Get Cost</li><li>• Get Users</li></ul>

## Preface

---

### About this Manual

This document is a user guide for the Cortex Amazon Web Interface Subtasks.

### Audience

The audience for this document is those wanting to understand how to use CTX-AWS module.

### Related Material

Document
CTX-AWS - Deployment Plan
CTX-AWS.studiopkg

### Abbreviations used in this Document

<b>AWS</b>	Amazon Web Services
<b>CW</b>	CloudWatch
<b>EBS</b>	Elastic Block Storage
<b>EC2</b>	Elastic Compute Cloud
<b>SNS</b>	Simple Notification Service
<b>VM</b>	Virtual Machine
<b>IAM</b>	Identity and Access Management
<b>ID</b>	Identification

## Requirements

---

The Cortex subtasks require the following:

- AWS Account Subscription
- Cortex PowerShell OCI
- PowerShell v5
- AWSPowerShell Module to be installed

Instructions for how to install these are included in the deployment plan.

## Integration

### Integration with Third-Party Systems

#### Amazon Web Services

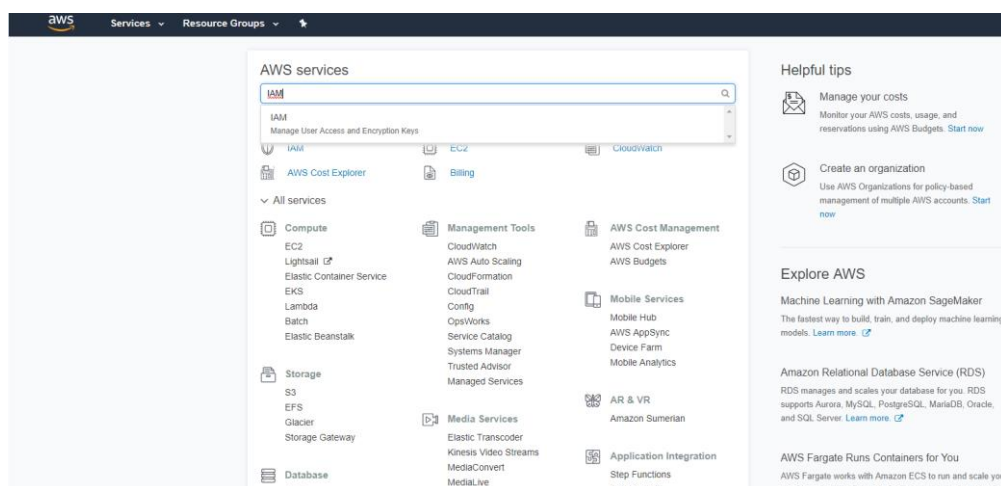
The Cortex AWS Subtasks will interact with the following third-party systems:

- Amazon Web Services systems

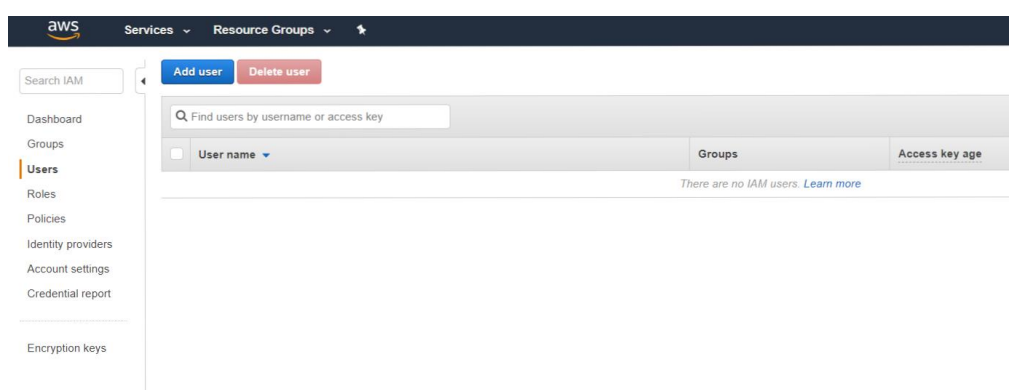
There is a need to configure at least one IAM user that has relevant authorisations for the CTX-AWS subtasks to be functional. In this example, one user will be configured with the relevant authorisations. It is possible to configure several users each with different permissions thereby giving a user authorisation to some functionality but not all.

#### Creating IAM Users

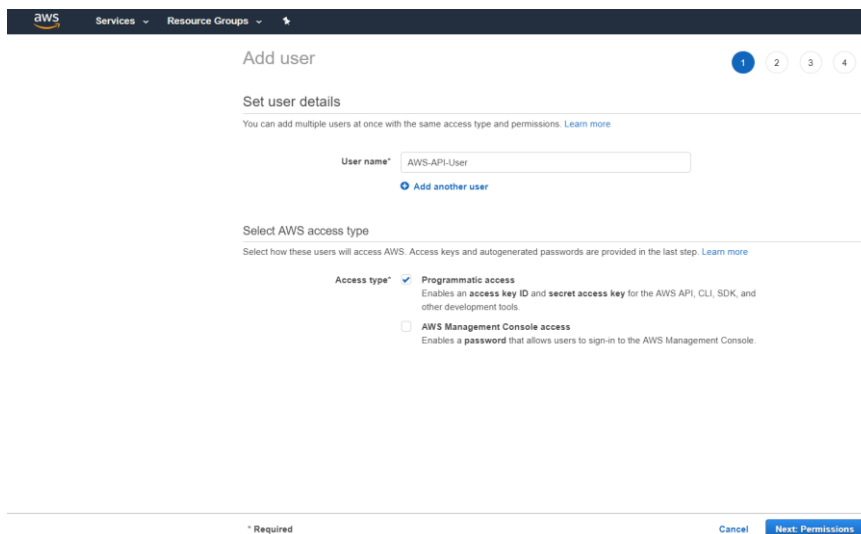
To create a new IAM user sign in to 'Amazon Management Console' and navigate to 'IAM Management Console'.



From here click 'Users' then 'Add user' to add a new user.



The name of the 'User name' needs to be specified and it is **paramount** that the user is given 'Programmatic Access'. Then click 'Next: Permissions'.



**Add user**

**Set user details**

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

[Add another user](#)

**Select AWS access type**

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

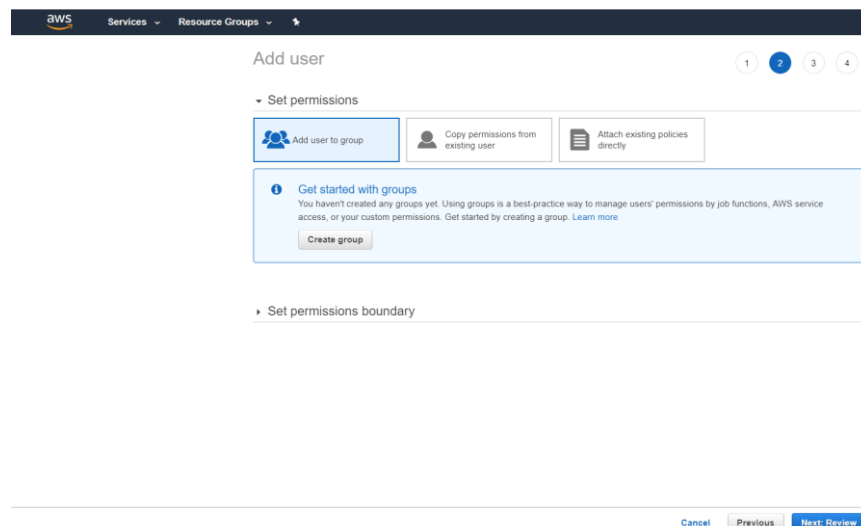
Access type\* ☒ **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☐ **AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

\* Required

[Cancel](#) [Next: Permissions](#)

Groups will be created later, and the user added to this group. For this reason, click 'Next: Review'.



**Add user**

**Set permissions**

[Add user to group](#) [Copy permissions from existing user](#) [Attach existing policies directly](#)

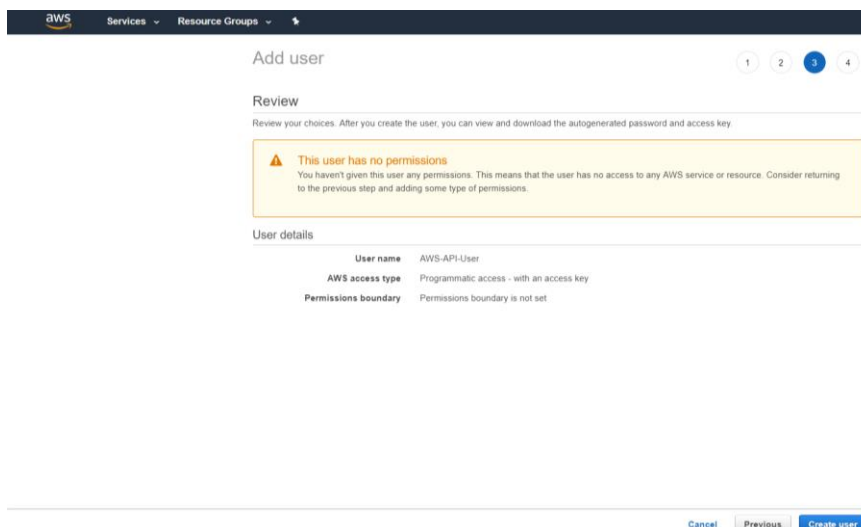
**Get started with groups**  
You haven't created any groups yet. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Get started by creating a group. [Learn more](#)

[Create group](#)

**Set permissions boundary**

[Cancel](#) [Previous](#) [Next: Review](#)

Then click 'Create User'.



**Add user**

**Review**

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

**This user has no permissions**  
You haven't given this user any permissions. This means that the user has no access to any AWS service or resource. Consider returning to the previous step and adding some type of permissions.

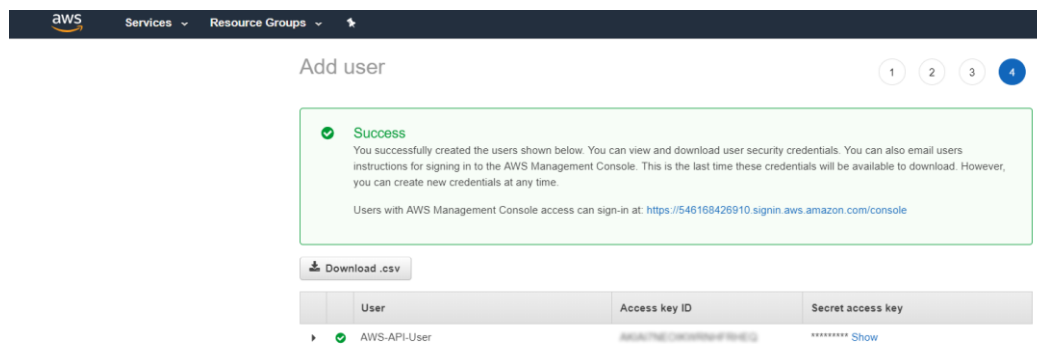
**User details**

User name	AWS-API-User
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

[Cancel](#) [Previous](#) [Create user](#)



The security credentials need to be downloaded once the user has been created as these cannot be accessed later. It is possible to download the credentials as a csv or they can be copied to location of choice.

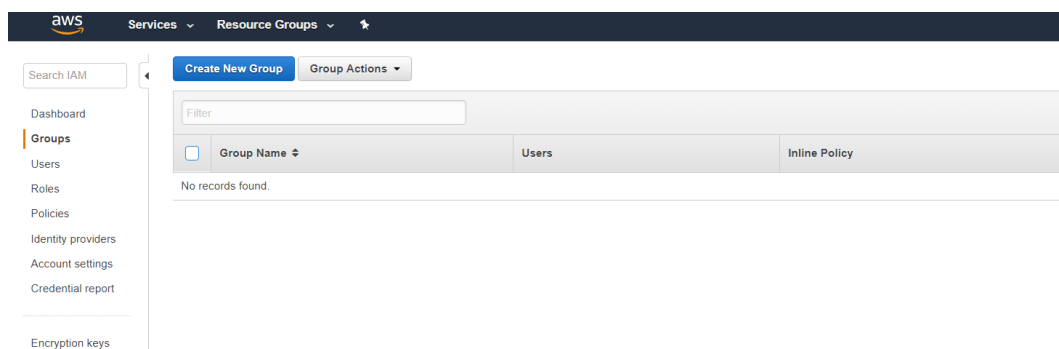


The screenshot shows the 'Add user' page in the AWS IAM console. A green success message states: 'You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time. Users with AWS Management Console access can sign-in at: <https://546168426910.signin.aws.amazon.com/console>'. Below the message is a 'Download .csv' button. A table lists the created users:

User	Access key ID	Secret access key
AWS-API-User	AKIAI2PNEU3K6R9W4P6E2Q	***** <a href="#">Show</a>

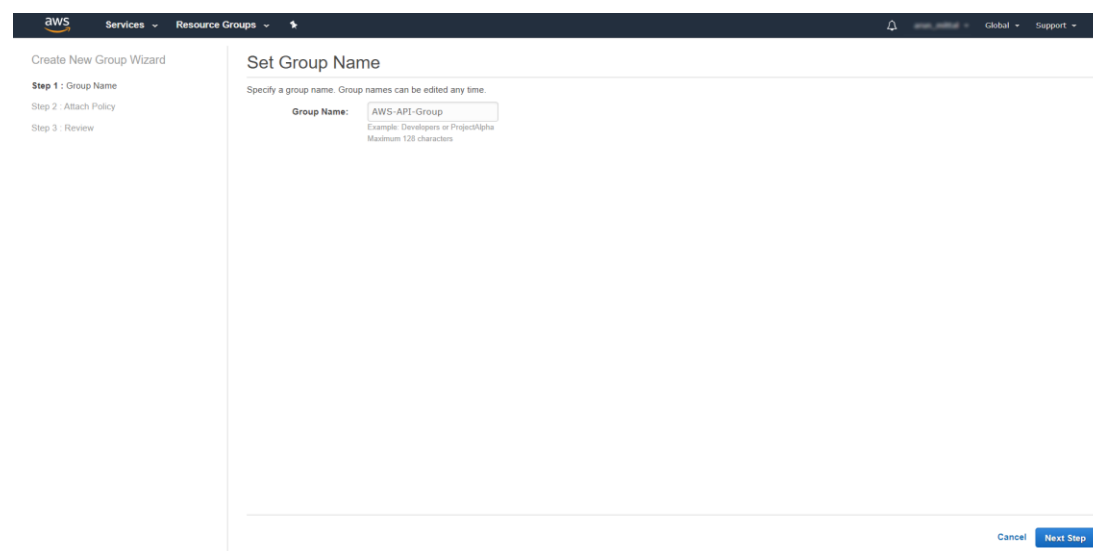
## Creating IAM Groups and Attaching Policies

To create an IAM group navigate to the 'IAM Management Console'. Then click 'Groups' then 'Create New Group'.



The screenshot shows the 'Create New Group' page in the AWS IAM console. The left sidebar contains a search bar and a list of IAM entities: Dashboard, Groups (selected), Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area has a 'Create New Group' button and a 'Group Actions' dropdown. Below is a table with columns for 'Group Name', 'Users', and 'Inline Policy'. The table is currently empty, showing 'No records found'.

Give the group an appropriate name and click 'Next Step'.




The screenshot shows the 'Set Group Name' page in the AWS IAM console. The left sidebar shows the 'Create New Group Wizard' with steps: Step 1: Group Name (selected), Step 2: Attach Policy, and Step 3: Review. The main content area has a heading 'Set Group Name' and a subheading 'Specify a group name. Group names can be edited any time.' Below this is a text input field for 'Group Name' with the value 'AWS-API-Group'. A note below the field says 'Example: Developers or ProjectAlpha' and 'Maximum 128 characters'. At the bottom right are 'Cancel' and 'Next Step' buttons.

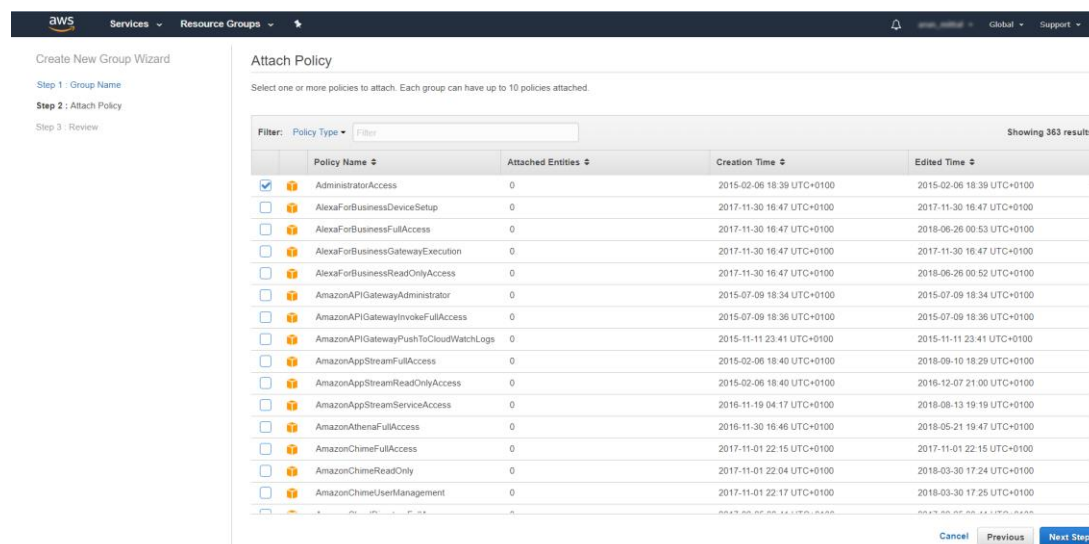
The following 'Policies' need to be attached to the group for the CTX-AWS subtasks to be functional:

- AdministratorAccess (Required for Cost and Usage otherwise this policy may be omitted)
- AmazonEC2FullAccess

- IAMFullAccess
- AmazonSNSFullAccess
- AmazonS3FullAccess

 **Note:** If AdministratorAccess has been added as a policy, then there is no need to attach any of the other policies.

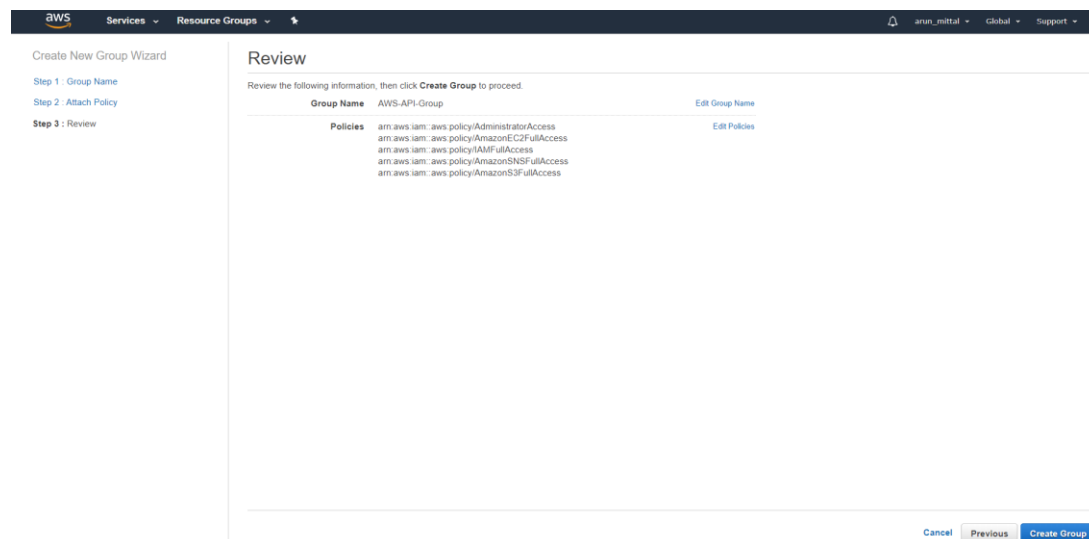
Then click 'Next Step'.



The screenshot shows the 'Attach Policy' step of the 'Create New Group Wizard'. The left sidebar shows the progress: Step 1: Group Name, Step 2: Attach Policy (current), and Step 3: Review. The main area is titled 'Attach Policy' and includes a filter bar and a table of available policies. The table has columns for Policy Name, Attached Entities, Creation Time, and Edited Time. The 'AdministratorAccess' policy is selected with a checkmark.

Policy Name	Attached Entities	Creation Time	Edited Time
<input checked="" type="checkbox"/> AdministratorAccess	0	2015-02-06 18:39 UTC+0100	2015-02-06 18:39 UTC+0100
<input type="checkbox"/> AlexaForBusinessDeviceSetup	0	2017-11-30 16:47 UTC+0100	2017-11-30 16:47 UTC+0100
<input type="checkbox"/> AlexaForBusinessFullAccess	0	2017-11-30 16:47 UTC+0100	2018-06-26 00:53 UTC+0100
<input type="checkbox"/> AlexaForBusinessGatewayExecution	0	2017-11-30 16:47 UTC+0100	2017-11-30 16:47 UTC+0100
<input type="checkbox"/> AlexaForBusinessReadOnlyAccess	0	2017-11-30 16:47 UTC+0100	2018-06-26 00:52 UTC+0100
<input type="checkbox"/> AmazonAPIGatewayAdministrator	0	2015-07-09 18:34 UTC+0100	2015-07-09 18:34 UTC+0100
<input type="checkbox"/> AmazonAPIGatewayInvokeFullAccess	0	2015-07-09 18:36 UTC+0100	2015-07-09 18:36 UTC+0100
<input type="checkbox"/> AmazonAPIGatewayPushToCloudWatchLogs	0	2015-11-11 23:41 UTC+0100	2015-11-11 23:41 UTC+0100
<input type="checkbox"/> AmazonAppStreamFullAccess	0	2015-02-06 18:40 UTC+0100	2018-09-10 18:29 UTC+0100
<input type="checkbox"/> AmazonAppStreamReadOnlyAccess	0	2015-02-06 18:40 UTC+0100	2016-12-07 21:00 UTC+0100
<input type="checkbox"/> AmazonAppStreamServiceAccess	0	2016-11-19 04:17 UTC+0100	2018-08-13 19:19 UTC+0100
<input type="checkbox"/> AmazonAthenaFullAccess	0	2016-11-30 16:46 UTC+0100	2018-05-21 19:47 UTC+0100
<input type="checkbox"/> AmazonChimeFullAccess	0	2017-11-01 22:15 UTC+0100	2017-11-01 22:15 UTC+0100
<input type="checkbox"/> AmazonChimeReadOnly	0	2017-11-01 22:04 UTC+0100	2018-03-30 17:24 UTC+0100
<input type="checkbox"/> AmazonChimeUserManagement	0	2017-11-01 22:17 UTC+0100	2018-03-30 17:25 UTC+0100

Then verify the policies listed above have been attached to the group, then click 'Create Group'.

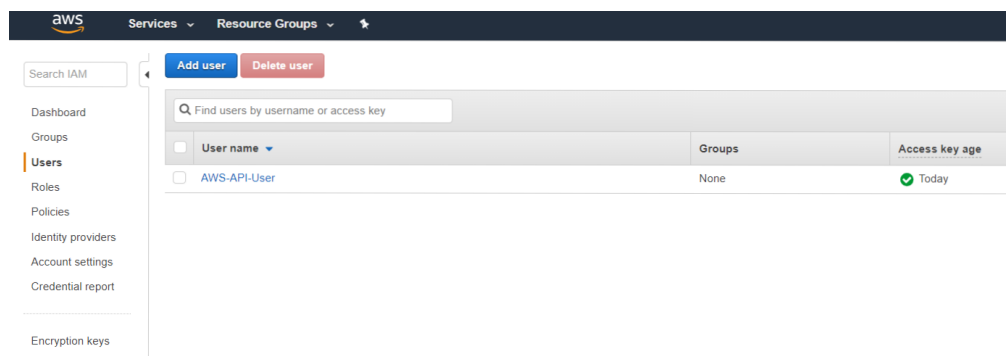


The screenshot shows the 'Review' step of the 'Create New Group Wizard'. The left sidebar shows the progress: Step 1: Group Name, Step 2: Attach Policy, and Step 3: Review (current). The main area is titled 'Review' and displays the group name 'AWS-API-Group' and the list of attached policies.

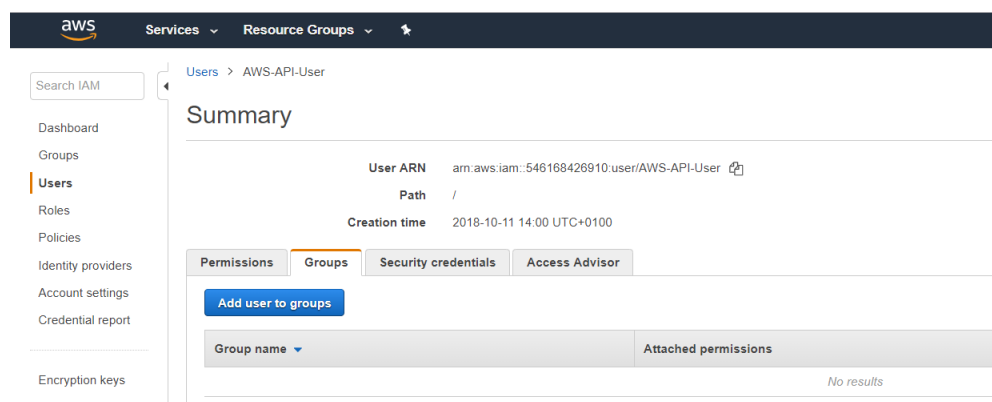
Group Name	Policies
AWS-API-Group	arn:aws:iam::aws:policy/AdministratorAccess arn:aws:iam::aws:policy/AmazonEC2FullAccess arn:aws:iam::aws:policy/IAMFullAccess arn:aws:iam::aws:policy/AmazonSNSFullAccess arn:aws:iam::aws:policy/AmazonS3FullAccess

## Adding Users to Groups

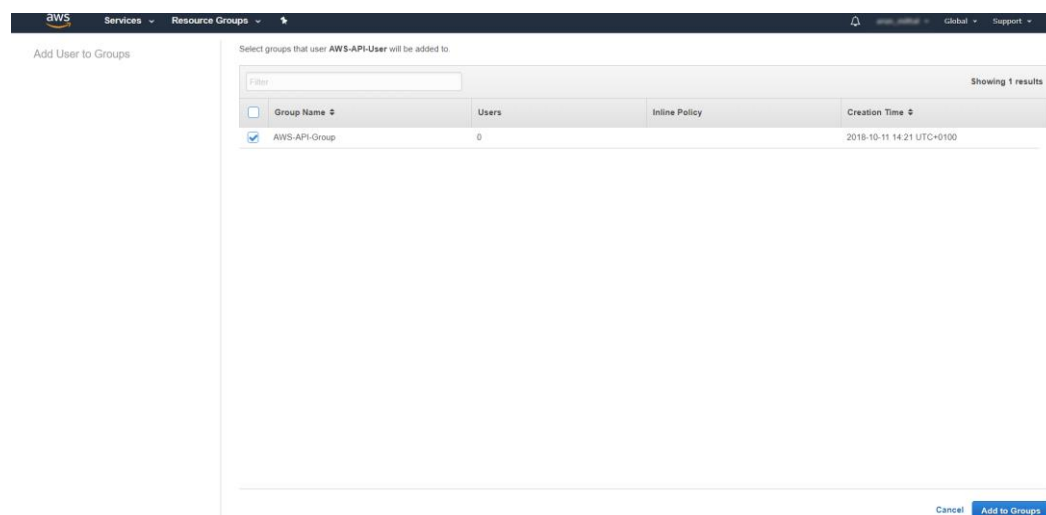
Users need to be added to the group that was created in Section 0. To do this, navigate to 'IAM Management Console' as shown in Section 0. Then click 'Users'.



Click on the user that was created previously, then click 'Groups' then click 'Add user to groups'.



From here select the group that was created previously and then click 'Add to Groups'.



The user should now have the relevant authorisation to perform actions using CTX-AWS subtasks.

## PowerShell

## Integrating with Existing Infrastructure

None Required.

## 1 AWS-EC2-CREATE-INSTANCE

---

### 1.1 Overview

This subtask creates an instance of server in a region.

### 1.2 Inputs

Input Variables	Type	Description
AECI_i_Instance-Type	Text	The type of instance, for the free tier “t2.micro” or “t2.nano”.
AECI_i_Key-Name	Text	The name of the keypair which will be used to logon to the VM.
AECI_i_Region	Text	The region where the instance is created. For Ireland “eu-west-1”, for London “eu-west-2”.
AECI_i_image-id	Text	The name of the image used to create the instance. E.g. ami-fbce3b9c
AECI_i_Credentials	Structure	Contains the elements access-key and secret-key for authentication.

### 1.3 Outputs

Output Variables	Type	Description
AECI_o_Instance-Id	Text	The Instance-Id of the instance created

## 2 AWS-EC2-GET-INSTANCE

---

### 2.1 Overview

This subtask gets the list of instances in a region.

### 2.2 Inputs

Input Variables	Type	Description
AEGI_i_Instance-Id (Optional)	Text	The instance id, if blank it gets all instances.
AEGI_i_Credentials	Structure	Contains the elements access-key and secret-key for authentication.

### 2.3 Outputs

Output Variables	Type	Description
AEGI_o_Instances	List	A list of all instances - instances are structures.

Example of an instance structure:

```
{
  "AMILAUNCHINDEX": 0,
  "ARCHITECTURE": {
    "VALUE": "x86_64"
  },
  "BLOCKDEVICEMAPPINGS": [
    "Amazon.EC2.Model.InstanceBlockDeviceMapping"
  ],
  "CLIENTTOKEN": null,
  "EBSOPTIMIZED": false,
  "ELASTICGPUASSOCIATIONS": [],
  "ENASUPPORT": true,
  "HYPERVISOR": {
    "VALUE": "xen"
  },
  "IAMINSTANCEPROFILE": {
    "ARN": "arn:aws:iam::708202619230:instance-profile/SytemManagerRole",
    "ID": "AIPAJMG2XFWSYWK7XWE76"
  },
  "IMAGEID": "ami-b8cd29df",
  "INSTANCEID": "i-053a7059a55509c88",
```

```

"INSTANCELIFECYCLE": null,
"INSTANCETYPE": {
  "VALUE": "t2.micro"
},
"KERNELID": null,
"KEYNAME": "WinSrv2012KeyPair",
"LAUNCHTIME": "2018-03-12 12:02:40",
"MONITORING": {
  "STATE": "disabled"
},
"NETWORKINTERFACES": [
  "Amazon.EC2.Model.InstanceNetworkInterface"
],
"PLACEMENT": {
  "AFFINITY": null,
  "AVAILABILITYZONE": "eu-west-2a",
  "GROUPNAME": null,
  "HOSTID": null,
  "SPREADDOMAIN": null,
  "TENANCY": "default"
},
"PLATFORM": {
  "VALUE": "Windows"
},
"PRIVATEDNSNAME": "ip-172-31-22-197.eu-west-2.compute.internal",
"PRIVATEIPADDRESS": "172.31.22.197",
"PRODUCTCODES": [],
"PUBLICDNSNAME": "ec2-52-56-244-217.eu-west-2.compute.amazonaws.com",
"PUBLICIPADDRESS": "52.56.244.217",
"RAMDISKID": null,
"ROOTDEVICENAME": "/dev/sda1",
"ROOTDEVICETYPE": {
  "VALUE": "ebs"
},
"SECURITYGROUPS": [
  "Amazon.EC2.Model.GroupIdentifier"
],
"SOURCEDESTCHECK": true,
"SPOTINSTANCEREQUESTID": null,
"SRIOVNETSUPPORT": null,

```

```
"STATE": {  
  "CODE": 16,  
  "NAME": "running"  
},  
"STATEREASON": null,  
"STATETRANSITIONREASON": null,  
"SUBNETID": "subnet-a3a1bcd8",  
"TAGS": [],  
"VIRTUALIZATIONTYPE": {  
  "VALUE": "hvm"  
},  
"VPCID": "vpc-9514fefb",  
"TAG": []  
}
```

## 3 AWS-EC2-START-INSTANCE

---

### 3.1 Overview

This subtask starts an instance in a region.

### 3.2 Inputs

Input Variables	Type	Description
AESI_i_Instance-Id	Text	The instance id, if blank it gets all instances.
AESI_i_Credentials	Structure	Contains the elements access-key and secret-key for authentication.

### 3.3 Outputs

Output Variables	Type	Description
AESI_o_State	Text	The state of the instance that has been started.



## 4 AWS-EC2-STOP-INSTANCE

---

### 4.1 Overview

This subtask stops an instance in a region.

### 4.2 Inputs

Input Variables	Type	Description
AESI_i_Instance-Id	Text	The instance id, if blank it gets all instances.
AESI_i_Credentials	Structure	Contains the elements access-key and secret-key for authentication.

### 4.3 Outputs

Output Variables	Type	Description
AESI_o_State	Text	The state of the instance that has been stopped.

## 5 AWS-EC2-RESTART-INSTANCE

---

### 5.1 Overview

This subtask restarts an instance in a region.

### 5.2 Inputs

Input Variables	Type	Description
AERI_i_Instance-Id	Text	The instance id, if blank it gets all instances.
AERI_i_Credentials	Structure	Contains the elements access-key and secret-key for authentication.

### 5.3 Outputs

None.

## 6 AWS-EC2-TERMINATE-INSTANCE

---

### 6.1 Overview

This subtask removes an instance from a region.

### 6.2 Inputs

Input Variables	Type	Description
AETI_i_Instance-Id	Text	The instance id, if blank it gets all instances.
AETI_i_Credentials	Structure	Contains the elements access-key and secret-key for authentication.

### 6.3 Outputs

Output Variables	Type	Description
AETI_O_State	Text	The state of the instance that has been terminated.

## 7 AWS-EC2-CHANGE-INSTANCE-TYPE

---

### 7.1 Overview

This subtask changes the instance type of an instance in a region.

### 7.2 Inputs

Input Variables	Type	Description
AECIT_i_Instance-Id	Text	The instance id, if blank it gets all instances.
AECIT_I_Type	Text	the new type of the instance, e.g t2.nano ( <a href="https://aws.amazon.com/ec2/instance-types/">https://aws.amazon.com/ec2/instance-types/</a> )
AECIT_i_Credentials	Structure	Contains the elements access-key and secret-key for authentication.

### 7.3 Outputs

Output Variables	Type	Description
AECIT_O_State	Text	The state of the instance that has been changed.

## 8 AWS-EC2-CREATE-CPU-ALARM-FOR-INSTANCE

### 8.1 Overview

This subtask sets up a CPU alarm for an instance in a region.

### 8.2 Inputs

Input Variables	Type	Description
AECCAFI_i_Alarm-Name	Text	Name of the alarm.
AECCAFI_i_Period-in-sec	Integer	Evaluation period time in seconds. Accepted values are: 60, 300 (5 mins), 900 (15 mins), 3600 (1 hour), 21600 (6 hours), 86400 (1 day).
AECCAFI_i_CPU-Threshold	integer	CPU percentatge to fire the alarm, e.g 70.
AECCAFI_i_Alarm-Action	Text	Notification topic, e.g. "arn:aws:sns:eu-west-2:708202619230:CPULAlarm"
AECCAFI_i_Instance-Id	Text	The instance id, if blank it gets all instances.
AECCAFI_i_Credentials	Structure	Contains the elements access-key and secret-key for authentication.

### 8.3 Check if CPU alarm is created

Alarm can be check in two places, the EC2 console and the CW console

CloudWatch console:

1. Open the CW console
2. Select Alarms on the left panel
3. Check if the new create alarm is there

EC2 console:

1. Open the EC2 console
2. Select Instances on the left panel
3. Select the Instance where the alarm was created
4. Click the Monitoring tab
5. Check that on the top of this tab there is a CloudWatch alarms section containing the alarm created.

## 9 AWS-CE-GET-COST

---

### 9.1 Overview

Retrieves cost and usage metrics for your account.

### 9.2 Inputs

Input Variables	Type	Description
ACGC_i_Start-Date	Text	The start date of the billing period.
ACGC_i_End-Date	Text	The end date of the billing period.
ACGC_i_Metric (optional)	Text	If the parameter is passed in, a detailed breakdown of cost will be provided.
ACGC_i_Granularity	Text	Can be either 'MONTHLY', or 'DAILY'
ACGC_i_Credentials	Structure	Contains the elements access-key and secret-key for authentication.

### 9.3 Outputs

Output Variables	Type	Description
ACGC_o_Usage	List	Returns the usage for each billing period

## 10 AWS-IAM-GET-USERS

---

### 10.1 Overview

Get all users on AWS account and their roles and policies.

### 10.2 Inputs

Input Variables	Type	Description
AIGU_i_Credentials	Structure	Contains the elements access-key and secret-key for authentication.

### 10.3 Outputs

Output Variables	Type	Description
AIGU_o_Users	List	List containing all users, their roles and policies associated with each role.