

Running Cloudify Manager on EC2 using the Cloudify shared AMI

Contents

[Why Cloudify shared AMI?](#)

[Prerequisites](#)

[Start Cloudify Shared AMI](#)

[Select the AMI ID](#)

[Select the Instance Type](#)

[Set Storage Size](#)

[Tag Instance](#)

[Configure Security Group](#)

[Launch the instance](#)

[Create Deployment for the CloudifySettings blueprint](#)

[Execute CloudifySettings Install workflow](#)

[Deploy the Hello World Blueprint](#)

[Place the pem file on the Manager Instance](#)

[Upload the Hello World Blueprint](#)

[Create the Hello Deployment](#)

[Install the Hello Deployment](#)

[Test the Hello Deployment](#)

[Shutdown and Restart](#)

[Terminating the running Blueprint](#)

[Stopping and Starting the Cloudify Manager AMI instance](#)

[Accessing the Instance on EC2](#)

[On Windows](#)

[On Linux](#)

[Install the Cloudify CLI](#)

[Problems?](#)

[Getting your AWS Access and Secret Keys](#)



If you are looking to experiment with Cloudify this is where you should begin!

This post will guide you how to start a Cloudify manager on EC2 using Cloudify Shared AMI and deploy hello world blueprint. Total execution time for this guide is **15 min**.

Why Cloudify shared AMI?

You may install and bootstrap Cloudify manager on any environment. This may take some time. The Cloudify Shared AMI will allow you to:

- Have a sandbox with unlimited resources to deploy blueprints that may span large number of nodes including multiple different regions world-wide.
- Benchmark and unit testing environment to measure Cloudify and your deployed applications ability to cope with large deployments/VMs. This is critical to have correct capacity planning for production environments.
- Have Cloudify manager available with (almost) Zero effort very quickly. No need to spend time installing and configuring it. If somehow you need another Cloudify manager machine, just hit the button and run another one on a brand new instance.

And...

- A great shared demo environment to brag about a new cool **open-source devops orchestration tool** you just learned about and showoff your TOSCA skills...

Prerequisites

Prior running a Cloudify shared AMI make sure you have:

1. EC2 account
2. Your pem file (private key file) and ppk file for windows users
3. Your EC2 account AWS access key ID
4. Your EC2 account secret access key
5. Your EC2 user/password

See below how to get these:

- <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html>
- [Getting your AWS Access and Secret Keys](#)
- <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>
- <http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSGettingStartedGuide/AWSCredentials.html>
- <http://tecadmin.net/convert-pem-to-ppk-private-key/>

Start Cloudify Shared AMI

Start Cloudify Shared AMI from here: http://getcloudify.org/thank_you_aws.htm

👍 Getting started with the Cloudify AWS AMI

Launch an AWS Pre-Baked Shared Image

US EAST AMI-EC9DD686

EU WEST 1 AMI-C8E1BA4

EU CENTRAL 1 AMI-3017095C

AP NORTH-EAST 1 AMI-28E4D746

AP SOUTHEAST 1 AMI-BADFC2D9

AP SOUTHEAST 2 AMI-E906C93A

Configuration

After your instance is up and running, the following procedure is needed in order to configure the newly running instance:

- ☐ Create a new deployment from the [CloudifySettings](#) blueprint. This blueprint will be pre-installed in your Cloudify Manager. You'll need to provide a number of inputs; see the **Configuration inputs** reference below for more info.
- ☐ Run the install workflow from the newly created deployment.
- ☐ Wait for the installation to be completed successfully. After installation is done, your manager should be in the same state as if it were just bootstrapped.

📖 Continue [reading the docs](#) for the full instructions on how to get your AMI up and running.

Select the AMI ID

Select the AMI ID (e.g. US EAST AMI-EC9DD686). This will take you to the EC2 control. You will have the log in if you haven't done so.



Sign In or Create an AWS Account

What is your e-mail or mobile number?

E-mail or mobile number:

☐ I am a new user.

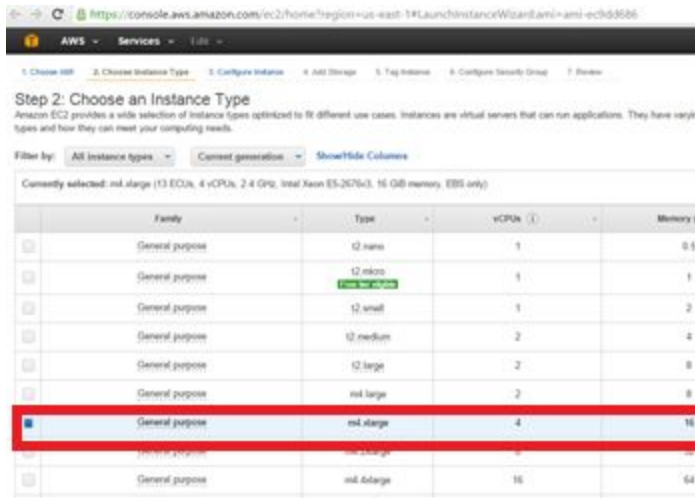
☒ I am a returning user
and my password is:

Sign in using our secure server

[Forgot your password?](#)

Select the Instance Type

Select the Instance Type - **m4.large** is minimal recommended instance type for the manager. Smaller Instance type with less vCPUs and memory will not work.



Configure the Instance details as usual.

Set Storage Size

Set the Storage size to 32 GB or larger:

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Tag Instance | 6. Configure Security Group | 7. Review

Step 4: Add Storage

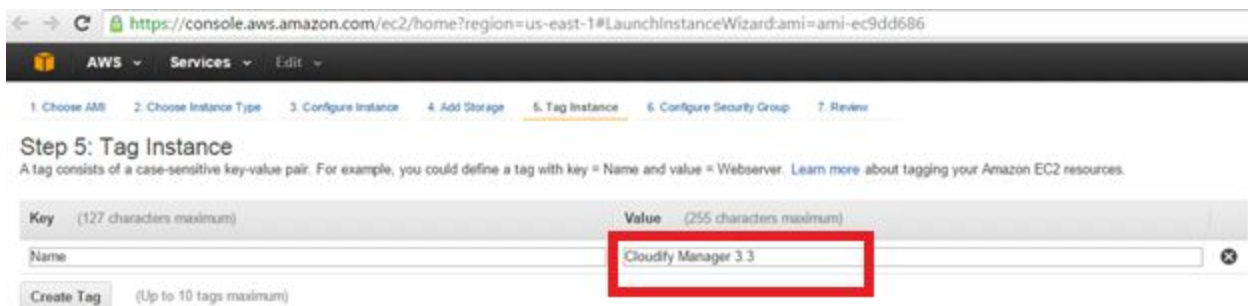
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type (1)	Device (1)	Snapshot (1)	Size (GiB) (1)	Volume Type (1)	IOPS (1)	Delete on Termination (1)	Encrypted (1)
Root	/dev/sda1	snap-e9dad1e7	32	General Purpose SSD (GP2)	96 / 3000	<input checked="" type="checkbox"/>	Not Encrypted
Add New Volume							

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Tag Instance

Provide the instance reasonable value for the name key:



Configure Security Group

On the Configure Security Group step - Make sure you have the right security group with relevant permissions. Here is the one I used:

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
All traffic	All	All	sg-78022a1c (default)
All traffic	All	All	sg-101a3294 (OpenVPN Access Server -HVM-2-0-17-AutogenByAWSMP-)

You will need SSH , HTTP access to the manager instance and other instances running the agent, so I suggest you have all ports available for all protocols available and later just the required ones. See below:

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
All traffic ▼	All	0 - 65535	Anywhere ▼ 0.0.0.0/0

Launch the instance

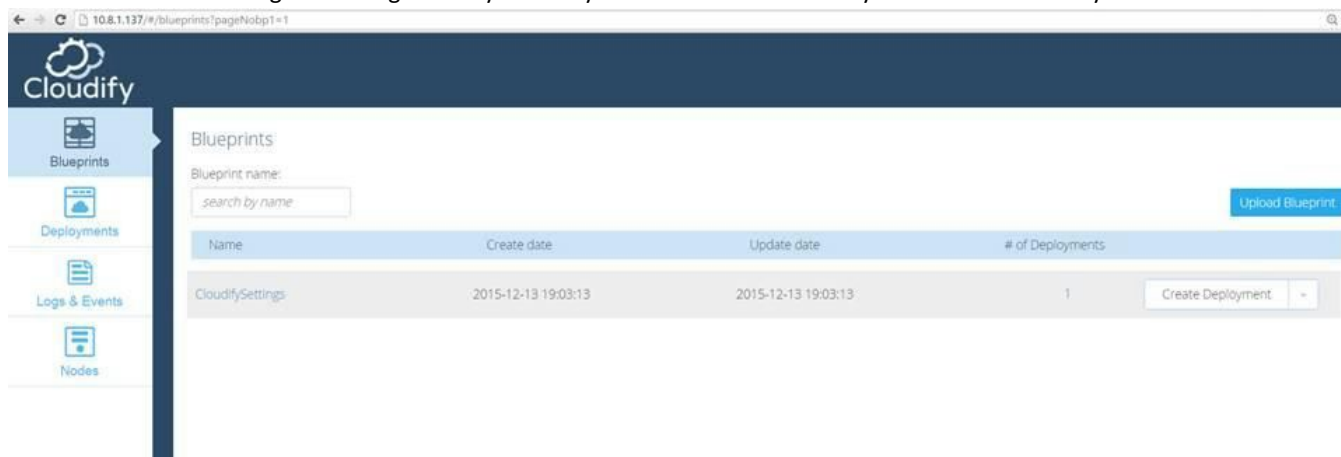
Before you launch the instance make sure you have the pem file (private key file) with you since you will need it later.

Review and Launch

Once you are done Click the button and launch the instance.

Create Deployment for the CloudifySettings blueprint

Once the instance running the manager is fully started you can access the Cloudify Web Console directly:



The screenshot shows the Cloudify Web Console interface. On the left is a sidebar with navigation links: Blueprints, Deployments, Logs & Events, and Nodes. The main content area is titled 'Blueprints' and includes a search bar labeled 'Blueprint name: search by name' and an 'Upload Blueprint' button. Below this is a table listing blueprints:

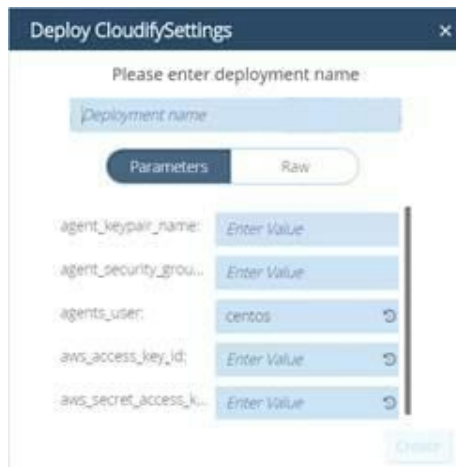
Name	Create date	Update date	# of Deployments
CloudifySettings	2015-12-13 19:03:13	2015-12-13 19:03:13	1

At the end of the 'CloudifySettings' row, there is a 'Create Deployment' button.

You should **Create Deployment** for the *CloudifySettings* blueprint provided by clicking the button.

Once clicked - You will see this:

Create Deployment



Deploy CloudifySettings

Please enter deployment name:

Deployment name

Parameters Raw

agent_keypair_name: Enter Value

agent_security_group_name: Enter Value

agents_user: centos

aws_access_key_id: Enter Value

aws_secret_access_key: Enter Value

Create

See example for values you should set for the *CloudifySettings* deployment:

Inputs:

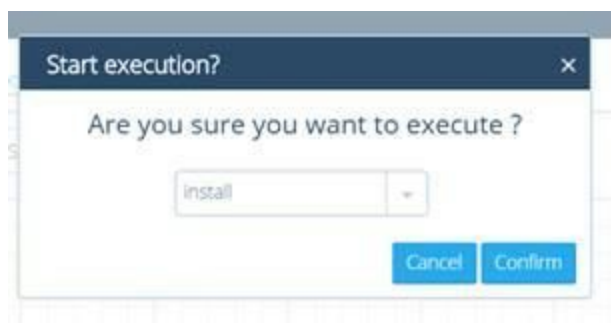
Input	Value
agent_keypair_name	shayCloudifySettings
agent_security_group_name	shayCloudifySettings-sg
agents_user	centos
aws_access_key_id	AKIAJ3634G62
aws_secret_access_key	UAUyX6o9RqqzoOeymRxZxsP8fd1Ea

The aws_access_key_id and aws_secret_access_key values can be retrieved from the [IAM console](#).

Make sure the agent_keypair_name agent_security_group_name you set **do not exists** as Cloudify will create these.

Execute CloudifySettings Install workflow

Once you deploy the CloudifySettings blueprint execute its **Install workflow**:





Start execution?

Are you sure you want to execute ?

Install

Cancel Confirm

You should see this once installed successfully:

Events		
	Workflow ended successfully	2016-01-18 14:24:00 'create_deployment_environment' workflow execution succeeded
	Task ended successfully	2016-01-18 14:24:00 Task succeeded 'riemann_controller.tasks.create'

When this is completed successfully your manager is ready to use.

Deploy the Hello World Blueprint

To test the manager you can deploy the hello world blueprint:

<https://github.com/cloudify-cosmo/cloudify-hello-world-example/tree/3.3>


Make sure you create blueprint for the [singlehost-blueprint.yaml](#).

Place the pem file on the Manager Instance

[Access the Manager machine](#). Copy the pem file to the manager instance. With windows you can use WinSCP. With linux you can use rcp.

Upload the Hello World Blueprint

Uploading via the Web UI:

Click the  button. The following dialog will be displayed.

Upload Blueprint

Select blueprint file to upload, and click 'Save'

* Select blueprint file or url

* Blueprint ID

Blueprint filename (e.g blueprint.yaml)

* Required field

Save

Fill in the values. See below example:

Upload Blueprint

Select blueprint file to upload, and click 'Save'

<https://github.com/cloudify-cosmo/cloudify-hello-wor>

hello

singlehost-blueprint.yaml

* Required field

Save

Uploading via the CLI:

Prior running this step see below [how to install the Cloudify CLI](#).

The following commands will get the blueprint from github (place it on the manager instance) and upload it:


```
sudo yum install unzip
```

```
wget https://github.com/cloudify-cosmo/cloudify-hello-world-example/archive/3.3.zip
```

```
unzip 3.3.zip
```

```
cfy blueprints upload -p cloudify-hello-world-example-3.3/singlehost-blueprint.yaml -b hello
```

Create the Hello Deployment

Create the Hello deployment via the Web Console by clicking the  for the **hello** blueprint. You will be prompt with the following:



The dialog box titled "Deploy hello" contains a close button (X) in the top right corner. Below the title, it says "Please enter deployment name". There is a text input field with the placeholder "Deployment name". Below this, there are two tabs: "Parameters" (selected) and "Raw". Under the "Parameters" tab, there are four input fields: "agent_private_key_p..." with placeholder "Enter Value", "agent_user:" with placeholder "Enter Value", "server_ip:" with placeholder "Enter Value", and "webserver_port:" with the value "8080" and a refresh icon. At the bottom right of the dialog is a "Create" button.

Here are examples for the values you should set:

Deployment name: hello

agent_private_key_path: /home/centos/mykeyfile.pem

agent_user: centos

server_ip: **Manager private IP**

webserver_port: 8080

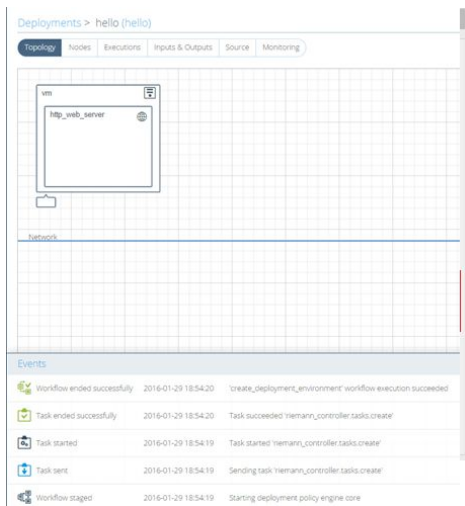
The above assumes you have copied your mykeyfile.pem file into /home/centos/ on the manager instance.

Please make sure you have the pem file in the right location on the manager machine as specified with the agent agent_private_key_path input property. Check the pem file permissions. It should have **read** permissions:

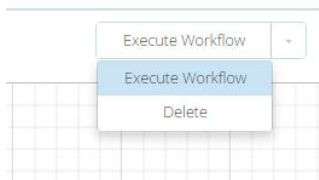
```
chmod 400 mykeyfile.pem
```

To access the manager instance follow the instructions at the [Accessing the Instance on EC2](#) section.

When you click the **Create button** the deployment process will start. Once completed you should see this:

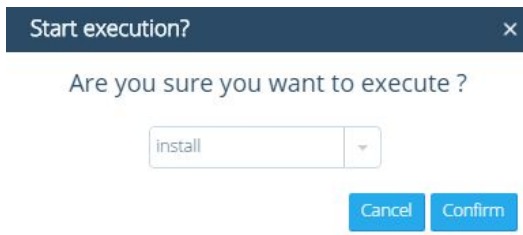


To install the blueprint click the **Execute Workflow** for the **hello** deployment:



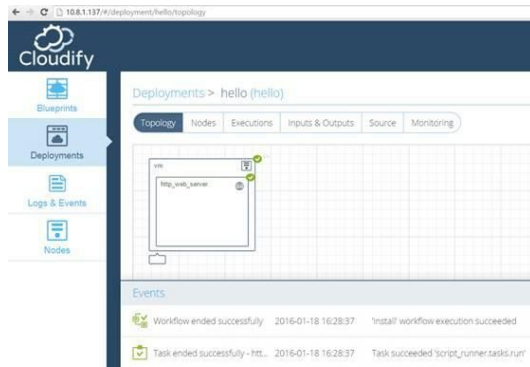
Install the Hello Deployment

Select install:



And Click **Confirm**.

Once it is successfully installed you should see this:



Test the Hello Deployment

If you have used the defaults you should see this when pointing your browser to the right URL:



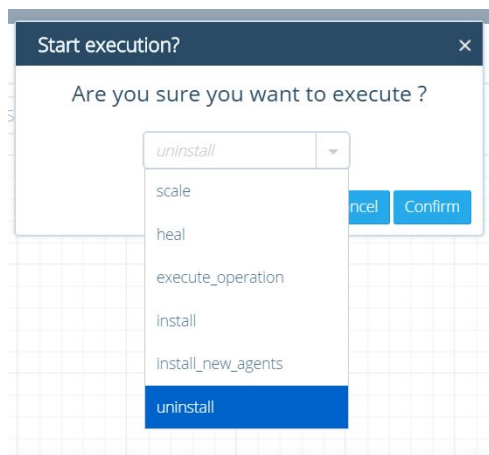
Congratulations: You have the Cloudify manager and a blueprint running!

Shutdown and Restart

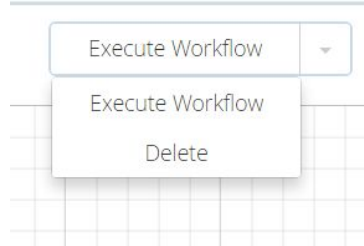
Terminating the running Blueprint

To terminate the running blueprint deployment:

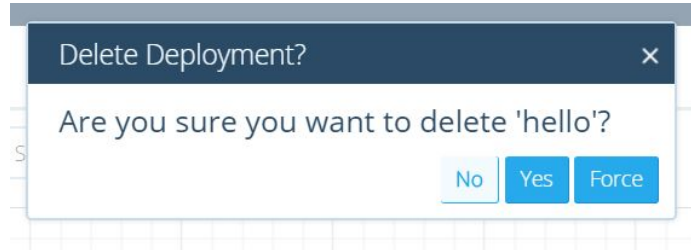
Select the Uninstall workflow for the Hello deployment:



Select the Delete the hello deployment:

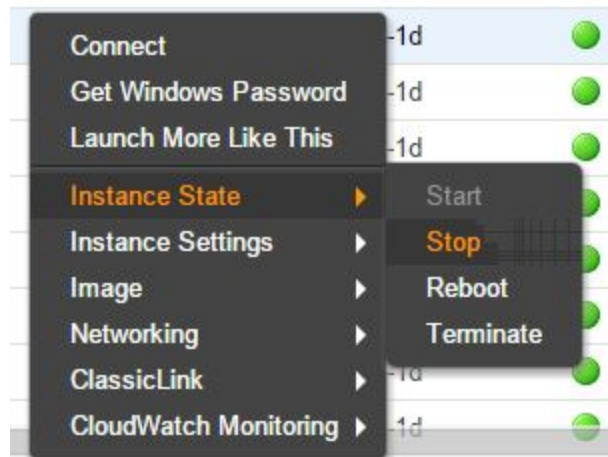


Delete the hello deployment:



Stopping and Starting the Cloudify Manager AMI instance

In this point you can stop the AMI instance running the Cloudify Manager- You can do this directly from EC2 console:



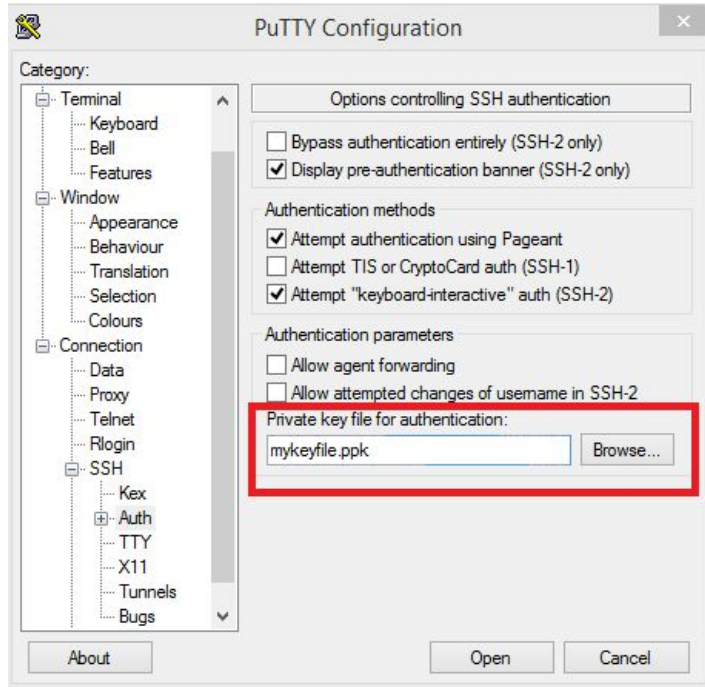
This will allow you later to start the instance and continue your work. To start the instance running the Cloudify manager:

03	Connect	us-east-1d
09	Get Windows Password	us-east-1d
09	Launch More Like This	us-east-1d
09	Instance State ▶	Start
09	Instance Settings ▶	Stop
09	Image ▶	Reboot
09	Networking ▶	Terminate
09	ClassicLink ▶	us-east-1d
01	CloudWatch Monitoring ▶	us-east-1d

Accessing the Instance on EC2

On Windows

Make sure you set the private key file:



You will need to generate the ppk file using [PuTTY Key Generator](#) from the pem file.

On Linux

`ssh -i mykeyfile.pem Manager public IP`

The user name to access the instance is **centos**.

Install the Cloudify CLI

To install the Cloudify CLI on the manager instance first access the instance using ssh / putty into.

Once you access the manager instance run the following:

```
sudo yum install wget
```

```
sudo wget
```

```
http://repository.cloudifysource.org/org/cloudify3/3.3.0/ga-RELEASE/cloudify-centos-Core-cli-3.3.0-ga\_b300.x86\_64.rpm
```

```
sudo rpm -Uvh cloudify-centos-Core-cli-3.3.0-ga_b300.x86_64.rpm
```

```
source /opt/cfy/env/bin/activate
```

```
cfy use -t <Manager private IP>
```

The last step should include the Manager private IP.

To verify the manager is running correctly run this:

cfy status

You should see this:

```
(env) [root@cloudify centos]# cfy status
Getting management services status... [ip=10.8.1.137]

Services:
+-----+-----+
| service | status |
+-----+-----+
| InfluxDB | running |
| Celery Management | running |
| Logstash | running |
| RabbitMQ | running |
| AMQP InfluxDB | running |
| Manager Rest-Service | running |
| Cloudify UI | running |
| Webserver | running |
| Riemann | running |
| Elasticsearch | running |
+-----+-----+
```

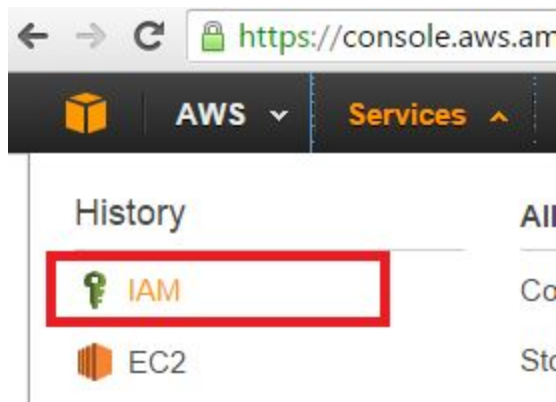
Problems?

If you have problems you probably have the wrong security group settings. Make sure all **Inbound** and **Outbound** traffic for all protocols and ports are available. Later you can limit this for open only relevant required protocols and ports. See:

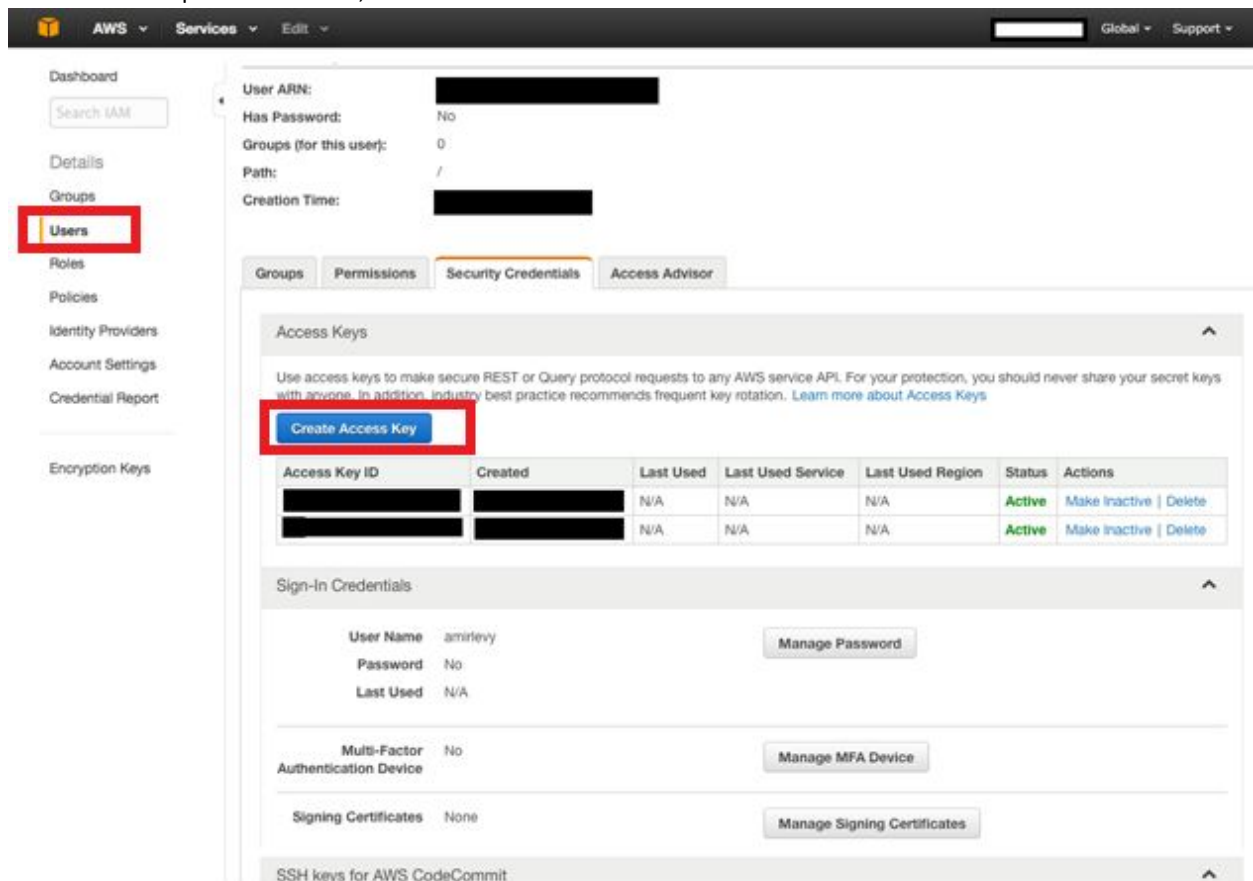
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html>

Getting your AWS Access and Secret Keys

To get your EC2 account AWS access key ID and your EC2 account secret access key access the IAM section under the Services within your EC2 console:



Click the Users Option on the left , Click the [Create Access Key](#) button:



The Create Access Key will be displayed. IT will show you your EC2 account AWS **access key ID** and your EC2 **account secret access key**:

Create Access Key

Your access key has been created successfully.

This is the last time these User security credentials will be available for download.

You can manage and recreate these credentials any time.

▼

Hide User Security Credentials

Access Key ID:

Secret Access Key:

Close

Download Credentials