

SELF-HOSTED APPLICATION

ALL-IN-ONE VERSION: GETTING STARTED GUIDE

Version 12.4

TABLE OF CONTENTS

| | |
|---|----|
| Introduction..... | 3 |
| Configure Services and Resources in AWS | 4 |
| Create an IAM Policy..... | 4 |
| Create an IAM User Group..... | 8 |
| Create an IAM User..... | 10 |
| Launch the Self-Hosted AMI..... | 14 |
| Configure the EC2 Instance | 16 |
| Install the Self-Hosted App | 22 |
| Configure the Self-Hosted App..... | 32 |
| License Your App..... | 32 |
| Create a Partner | 34 |
| Complete the Back-End System Configuration..... | 37 |
| Create a Trusted User..... | 39 |
| Create an Account..... | 41 |
| Create an IAM Role for Cross-Account Access..... | 44 |
| Create Least Privilege Policies | 50 |
| Attach Least Privilege Policies to Cross-Account Role | 53 |
| Upgrade the Self-Hosted App | 54 |
| Required Information | 57 |
| Frequently Asked Questions | 59 |
| Is There an Alternative to Remote Desktop?..... | 59 |
| Why Can't I Open My Browser?..... | 61 |
| Where Is My D: Drive? | 62 |
| How Do I Access My Log Files? | 63 |
| Appendix..... | 65 |
| IAM Policies..... | 65 |

Introduction

This document describes how to configure the All-In-One (AIO) self-hosted application.

To create a self-hosted environment where your data and security is completely protected, you will use an Amazon Machine Image (AMI) to configure an Amazon Elastic Compute Cloud (EC2) instance in your own virtual private cloud (VPC) that will contain all the application components:

- **Web Console:** how you will log into and use the application
- **Scheduler and workers:** the background processes that collect and store your AWS data
- **RDS database:** a Microsoft SQL[®] server database where your data gets stored
- **IAM role:** allows you to connect to your AWS account(s)

Note: These instructions require you to record key information generated from your configuration in AWS. For your convenience, use the [Required Information](#) section at the back of this document to record the data. Items you may wish to copy are highlighted in **yellow**.

Configure Services and Resources in AWS

Before the self-hosted application can access your AWS accounts, you need to create AWS credentials.

While a cross-account role is the preferred method for creating AWS credentials, the self-hosted application requires that you first create AWS Identity and Access Management (IAM) users as part of your back-end configuration. AWS will generate a unique access key and secret key for each IAM user. When you plug these keys into the Application-wide Configuration page, you enable the self-hosted application to collect the latest AWS pricing data.

This section will show you how to:

- create an IAM policy that will allow the self-hosted application to access AWS pricing data
- create an IAM user group and attach the pricing policy
- create an IAM user and add them to your user group

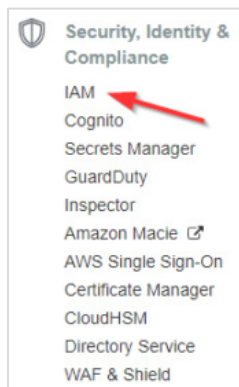
As a best practice, we recommend that you complete these steps in the prescribed order.

Note: To ensure that your self-hosted application contains a good cross-section of availability zones and pricing data, you must create each IAM user in a separate AWS account.

Create an IAM Policy

In this procedure, you will create an IAM policy that will allow the self-hosted application to access the AWS pricing data.

1. Launch the AWS Management Console associated with your first AWS account.
2. From the AWS Services page, scroll down to the Security, Identity & Compliance section and select **IAM**.



The Welcome to Identity and Access Management screen displays.

The screenshot shows the AWS IAM console's welcome page. At the top, it says 'Welcome to Identity and Access Management'. Below that, it provides an IAM users sign-in link: <https://cloudcheckrdev.signin.aws.amazon.com/console>. There is a 'Customize' link to the right. The 'IAM Resources' section shows: Users: 164, Roles: 310, Groups: 81, and Identity Providers: 0. Below that, 'Customer Managed Policies: 318' is listed. The 'Security Status' section shows a progress bar for '4 out of 5 complete'. A list of security tasks follows:

| Icon | Task | Action |
|------|-----------------------------------|--------|
| ⚠️ | Activate MFA on your root account | ▼ |
| ✅ | Create individual IAM users | ▼ |
| ✅ | Use groups to assign permissions | ▼ |
| ✅ | Apply an IAM password policy | ▼ |
| ✅ | Rotate your access keys | ▼ |

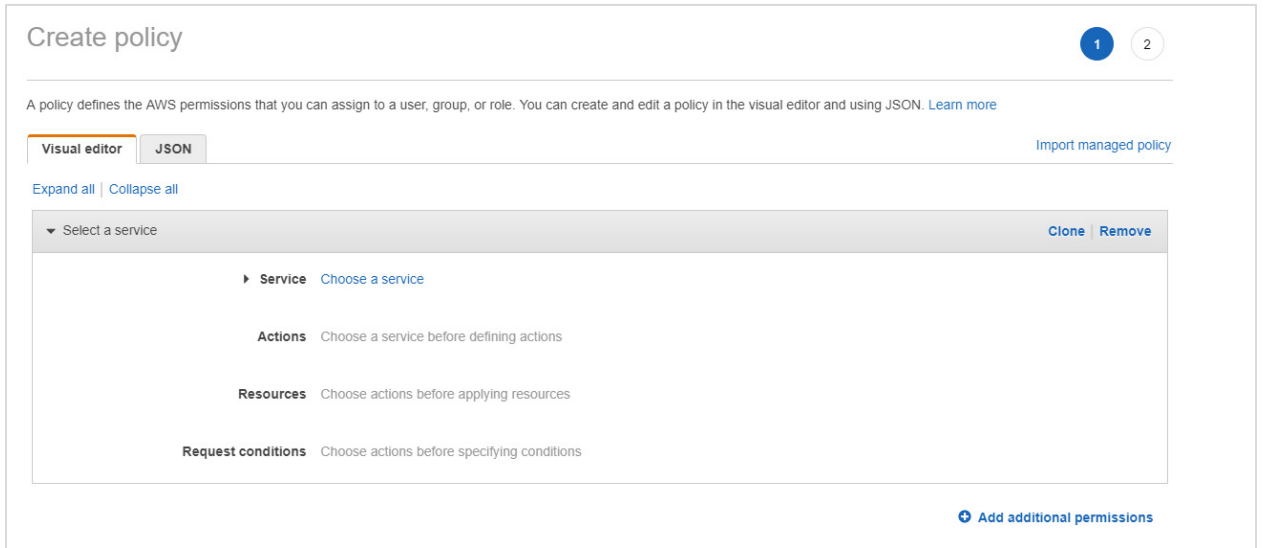
3. From the dashboard, click **Policies**.

The screenshot shows the navigation menu in the AWS IAM console. The menu items are: Dashboard, Groups, Users, Roles, **Policies** (highlighted with a red arrow), Identity providers, Account settings, and Credential report.

A list of policies displays.

4. Click **Create policy**.

5. The Create Policy page opens.



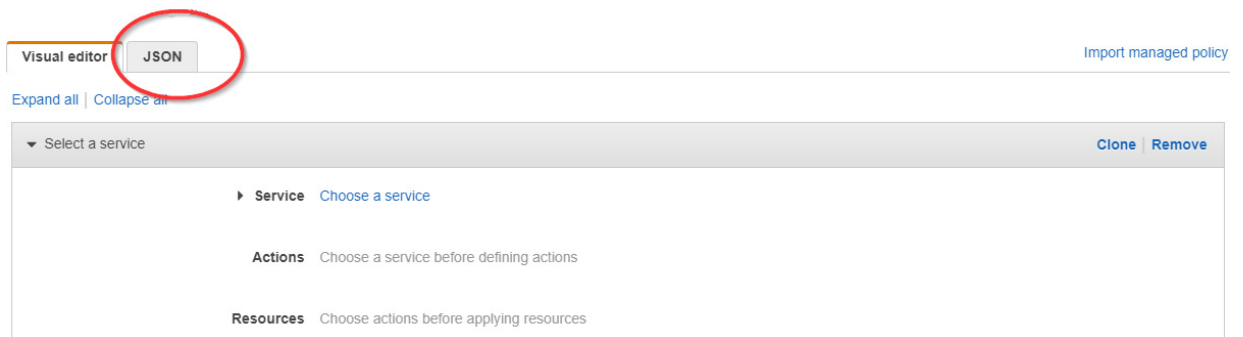
6. Follow the example in this step to see how to create the pricing policy:

a. Copy the permissions for the Pricing policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1470231538000",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSpotPriceHistory",
        "ec2:DescribeAvailabilityZones"
      ],
      "Resource": "*"
    }
  ]
}
```

b. Return to the Create Policy page in the AWS Management Console.

c. Click the **JSON** tab.



The JSON tab opens, allowing you to create a policy using JSON syntax.

- d. Replace the text in the JSON tab with the policy you just copied.
- e. Click **Review policy**. The Review policy page opens.
- f. Type a name for the policy and click **Create policy**.

Create policy 1 2

Review policy

Name* ←
Use alphanumeric and '+', '@', '_' characters. Maximum 128 characters.

Description
Maximum 1000 characters. Use alphanumeric and '+', '@', '_' characters.

Summary

| Service | Access level | Resource | Request condition |
|--|---------------|---------------|-------------------|
| Allow (1 of 171 services) Show remaining 170 | | | |
| EC2 | Limited: List | All resources | None |

* Required Cancel Previous **Create policy** ←

A message indicates that AWS has created your policy.

✔ PricingPolicy has been created. ← ✕

Create policy Policy actions ↻ ⚙️ ⓘ

Filter policies Showing 2 results

| | Policy name | Type | Used as | Description |
|-----------------------|----------------------|------------------|---------|-------------|
| <input type="radio"/> | pricing_policy | Customer managed | None | |
| <input type="radio"/> | PricingPolicy | Customer managed | None | |

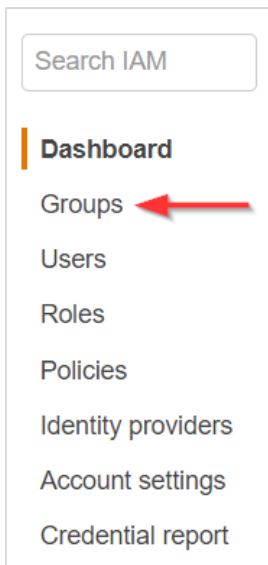
7. Repeat steps 1-6 to create a policy for the remaining two AWS accounts.
8. Copy the pricing policy name to the **Required Information** section.

Create an IAM User Group

An IAM user group allows you to apply group permissions to all users in that group automatically.

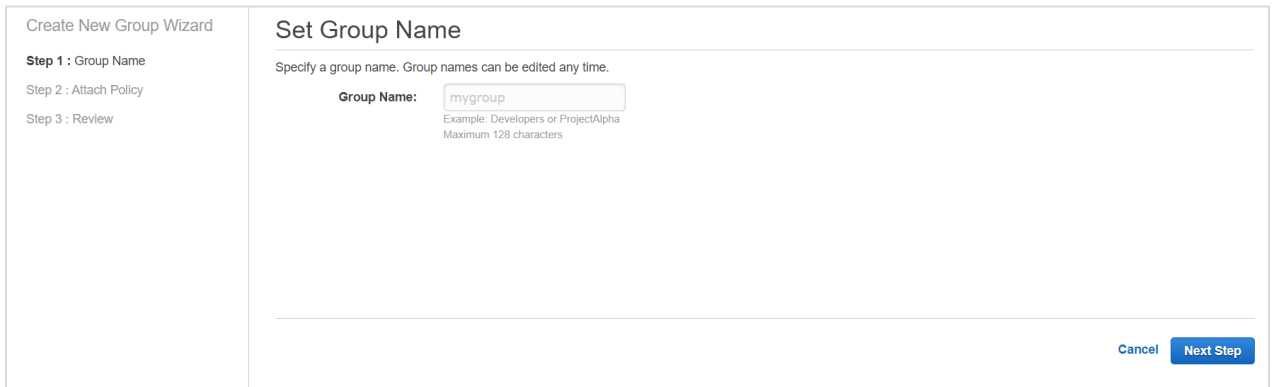
In this procedure, you will create an IAM user group and attach the policy you created in the [Create an IAM Policy](#) section.

1. From the AWS Services page, scroll down to the Security, Identity & Compliance section and select **IAM**.
The Welcome to Identity and Access Management screen displays.
2. From the dashboard, click **Groups**.



A list of groups displays.

3. Click **Create New Group**. The Create New Group wizard opens.

A screenshot of the "Set Group Name" step in the "Create New Group Wizard". The wizard progress bar shows "Step 1: Group Name" as the current step, followed by "Step 2: Attach Policy" and "Step 3: Review". The main content area is titled "Set Group Name" and includes the instruction "Specify a group name. Group names can be edited any time." Below this is a "Group Name:" label followed by a text input field containing "mygroup". Underneath the input field, there is an example: "Example: Developers or ProjectAlpha" and a note: "Maximum 128 characters". At the bottom right of the form, there are "Cancel" and "Next Step" buttons.

4. Type a group name. Click **Next Step**.

A list of policies displays.

5. Select the checkbox next to the pricing policy that you configured in the [Create an IAM Policy](#) section.
6. Click **Next Step**.

The page displays your group name and the attached policies. For the purposes of this procedure, we named the new IAM group **PricingGroup**.

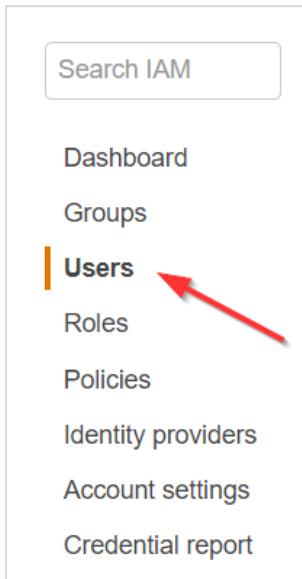
The screenshot shows the 'Review' step of the 'Create New Group Wizard'. On the left, a sidebar lists the steps: 'Step 1 : Group Name', 'Step 2 : Attach Policy', and 'Step 3 : Review'. The main area is titled 'Review' and contains the text: 'Review the following information, then click **Create Group** to proceed.' Below this, there are two rows of information. The first row is 'Group Name' with the value 'PricingGroup' and a red arrow pointing to it, and an 'Edit Group Name' link. The second row is 'Policies' with the value 'arn:aws:iam::106491416295:policy/PricingPolicy' and a red arrow pointing to it, and an 'Edit Policies' link. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Create Group' (which is highlighted in blue).

7. Click **Create Group**.
AWS adds your new IAM user group to the list.
8. Repeat steps 1-7 to create IAM user groups for the remaining two AWS accounts.
9. Copy the IAM group names to the [Required Information](#) section.

Create an IAM User

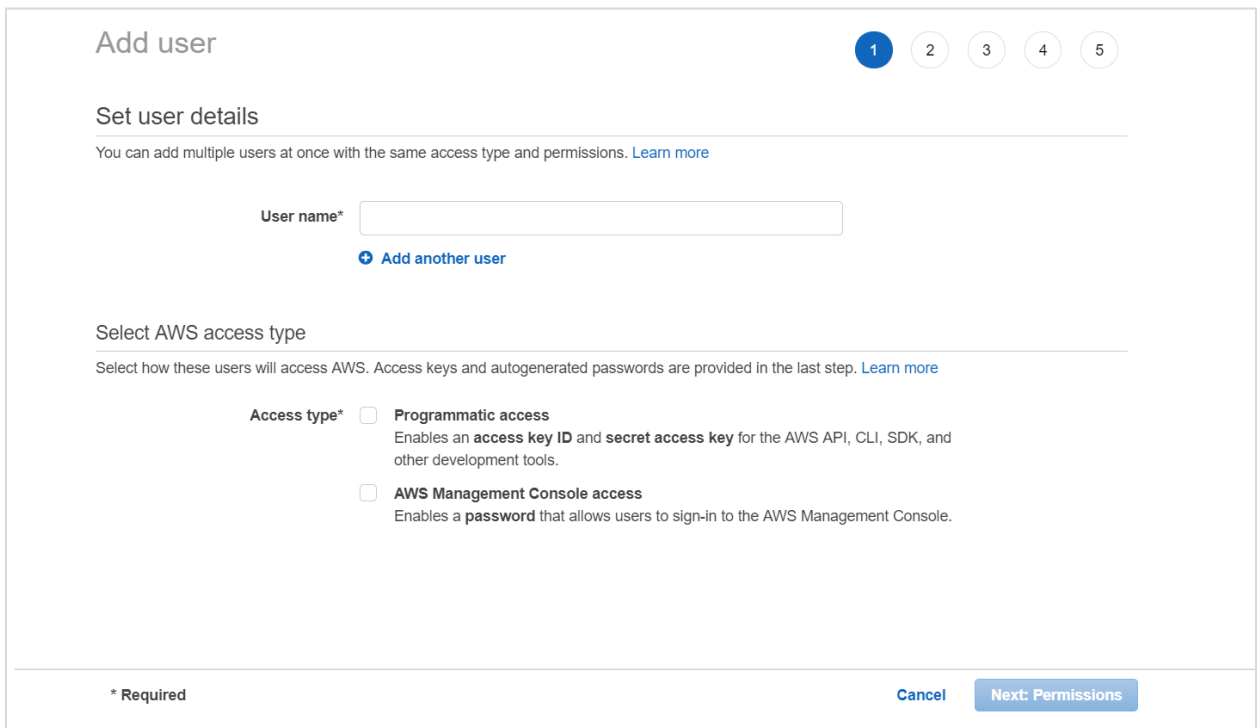
This procedure will show you how to create an IAM user in AWS.

1. From the AWS Services page, scroll down to the Security, Identity & Compliance section and select **IAM**.
The Welcome to Identity and Access Management screen displays.
2. From the dashboard, click **Users**.



A list of users displays.

3. Click **Add user**. The Add User wizard opens.

A screenshot of the AWS IAM "Add user" wizard. The title "Add user" is at the top left, and a progress indicator with five steps (1-5) is at the top right, with step 1 highlighted. The section "Set user details" is active. Below it is a text input field for "User name*" and a blue link "+ Add another user". The section "Select AWS access type" is below that. It contains two radio button options: "Programmatic access" (selected) and "AWS Management Console access". The "Programmatic access" option has a description: "Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools." The "AWS Management Console access" option has a description: "Enables a password that allows users to sign-in to the AWS Management Console." At the bottom left is the text "* Required". At the bottom right are two buttons: "Cancel" and "Next: Permissions".

4. On this screen:

- Type a username.
- Select the **Programmatic access** check box so you can generate access and secret keys.
- Click **Next: Permissions**.

Add user 1 2 3 4 5

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

User name* pricing_user

[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**

The Set permissions screen displays. The Add user to group button is selected by default.

5. Select the checkbox next to the group you created in the Create an IAM User Group section and click **Next: Tags**.

Add user 1 2 3 4 5

▼ **Set permissions**

Add user to group Copy permissions from existing user Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Add user to group

Create group Refresh

Q Search Showing 4 results

| Group | Attached policies |
|--|-------------------|
| <input type="checkbox"/> CloudCheckrTesters | CloudCheckrTester |
| <input checked="" type="checkbox"/> PricingGroup | PricingPolicy |
| <input type="checkbox"/> ReadOnly | ReadOnlyAccess |
| <input type="checkbox"/> SelfHostedGroup | None |

Cancel Previous **Next: Tags**

The Add tags page, which is optional, displays. For the purposes of this procedure, we will not add tags.

6. Click **Next: Review**.

This page displays the name of the user.

Add user 1 2 3 4 5

Review

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

User details

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

Permissions summary

The user shown above will be added to the following groups.

| Type | Name |
|-------|--------------|
| Group | PricingGroup |

Tags

No tags were added.

Cancel Previous **Create user**

7. Click **Create user**.

A message lets you know that AWS created the user and the associated access and secret keys.

Add user 1 2 3 4 5

Success

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://ccselfhosted.signin.aws.amazon.com/console>

Download .csv

| User | Access key ID | Secret access key |
|--------------|--------------------------------------|-------------------|
| pricing_user | AKIAI44QH8DHBVS7GALP3C4V4VVD3NULBY4E | ***** Show |

Close

8. Click **Download .csv** to save the keys to a secure location and click **Close**.

Note: This is the only time these unique values are available for download or to copy. However, if you misplace them, you or your administrator can create new access and secret keys. See this [AWS topic](#) for more details.

9. Repeat steps 1-8 for the remaining two AWS accounts.

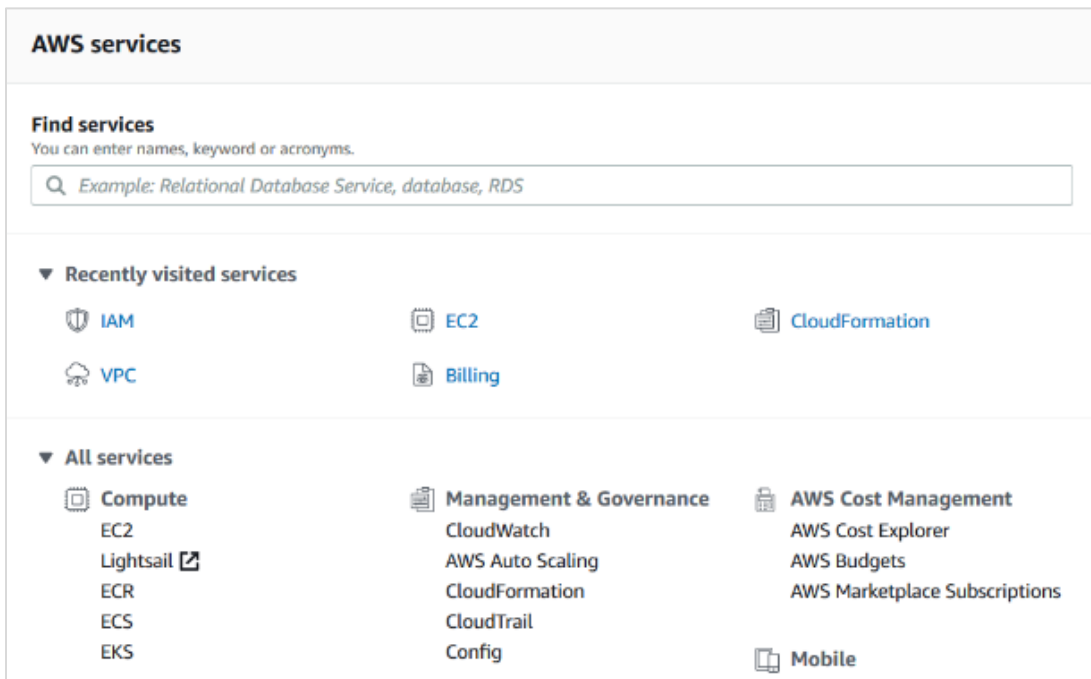
10. For each of the three IAM users, copy the username, access key, and secret key to the [Required Information](#) section.

Launch the Self-Hosted AMI

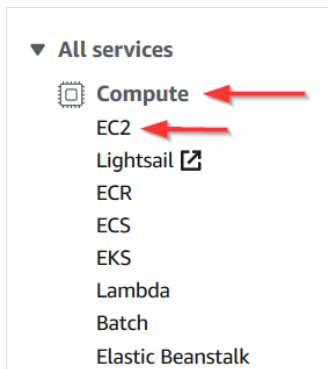
Once your sales representative has informed you that your AMI is available, you will need to launch it in AWS to begin the configuration process.

1. Log into the AWS Management Console.

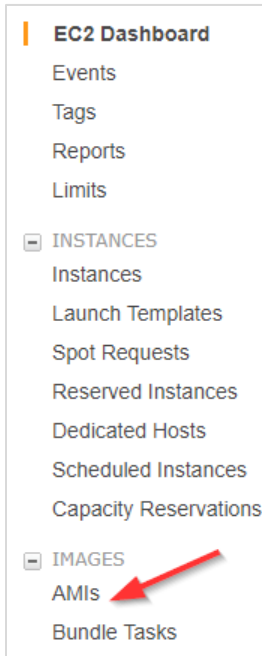
The AWS services page opens.



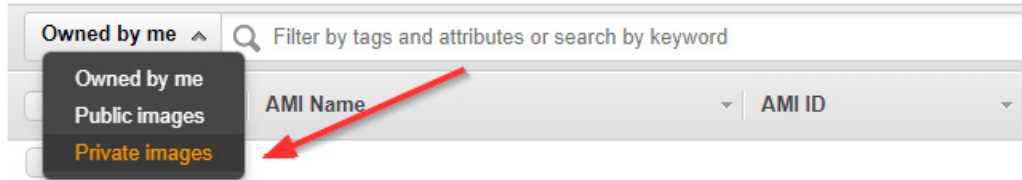
2. From the middle of the page, choose **Compute > EC2**.



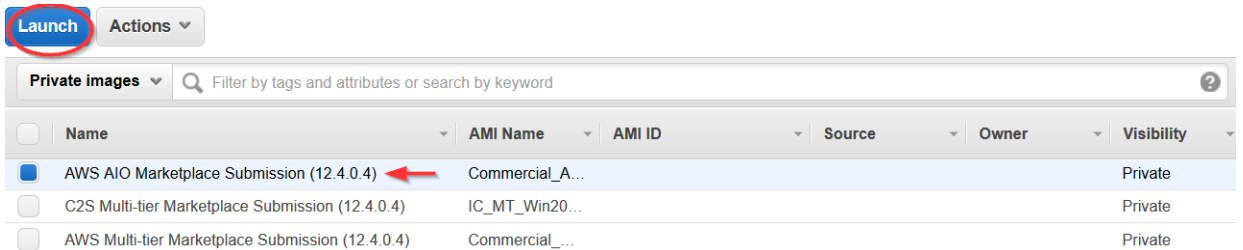
- From the EC2 Dashboard, select **Images > AMIs**.



- Click the drop-down arrow next to **Owned by me** and select **Private images**.



- Select the radio button next to the private offering identified by your sales representative and click **Launch**.



Configure the EC2 Instance

After you click **Launch**, AWS opens a template where you can configure the software settings for your EC2 instance.

Since you already purchased the AMI, Step 1, **Choose AMI**, is complete. Step 2: **Choose an Instance Type** is where you choose the type of EC2 instance, which is the virtual server where the self-hosted application will run.

Step 2: Choose an Instance Type
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: **All instance types** **Current generation** **Show/Hide Columns**

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

| | Family | Type | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance | IPv6 Support |
|-------------------------------------|-----------------|---|-------|--------------|-----------------------|-------------------------|---------------------|--------------|
| <input type="checkbox"/> | General purpose | t2.nano | 1 | 0.5 | EBS only | - | Low to Moderate | Yes |
| <input checked="" type="checkbox"/> | General purpose | t2.micro <small>Free tier eligible</small> | 1 | 1 | EBS only | - | Low to Moderate | Yes |
| <input type="checkbox"/> | General purpose | t2.small | 1 | 2 | EBS only | - | Low to Moderate | Yes |
| <input type="checkbox"/> | General purpose | t2.medium | 2 | 4 | EBS only | - | Low to Moderate | Yes |
| <input type="checkbox"/> | General purpose | t2.large | 2 | 8 | EBS only | - | Low to Moderate | Yes |
| <input type="checkbox"/> | General purpose | t2.xlarge | 4 | 16 | EBS only | - | Moderate | Yes |
| <input type="checkbox"/> | General purpose | t2.xlarge | 8 | 32 | EBS only | - | Moderate | Yes |
| <input type="checkbox"/> | General purpose | t3.nano | 2 | 0.5 | EBS only | Yes | Up to 5 Gigabit | Yes |
| <input type="checkbox"/> | General purpose | t3.micro | 2 | 1 | EBS only | Yes | Up to 5 Gigaabit | Yes |

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

1. Select the checkbox next to **c5.large**.

Note: This suggestion is designed to keep your costs down without sacrificing any performance in the self-hosted product. However, you must continue to monitor the cost and performance of your EC2 instance and modify your instance type to determine what works best in your deployment.

2. Click **Next: Configure Instance Details**.

Step 3: Configure Instance Details is where you configure your software and network requirements.

The screenshot shows the 'Step 3: Configure Instance Details' page in the AWS console. The page is divided into several sections, each with a label and a value. The 'Number of instances' is set to 1. The 'Purchasing option' is 'Request Spot instances'. The 'Network' is set to 'vpc-f41b3a8f (default)'. The 'Subnet' is set to 'No preference (default subnet in any Availability Zone)'. The 'Auto-assign Public IP' is set to 'Use subnet setting (Enable)'. The 'Placement group' is 'Add instance to placement group'. The 'Capacity Reservation' is set to 'Open'. The 'Domain join directory' is 'No directory'. The 'IAM role' is 'None'. The 'CPU options' are 'Specify CPU options'. The 'Shutdown behavior' is set to 'Stop'. The 'Enable termination protection' is 'Protect against accidental termination'. At the bottom right, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Storage'.

3. At a minimum, configure the following settings:

| Option | Description | Action |
|---------|--|---|
| Network | An isolated virtual network dedicated to your AWS account where you will launch your EC2 instance. | Select the network that contains your virtual private cloud (VPC). |
| Subnet | A range of IP addresses in your VPC | When you select a VPC, the subnet associated with that VPC will auto-populate this field. |

For the configuration of the self-hosted version, you can leave all other settings in their default state.

4. Click **Next: Add Storage**.

Step 4: Add Storage displays. This step is where you verify that you have the right volume and device, which make up the D: drive for your self-hosted application.

5. Verify the settings in their default state and modify as necessary:

- Volume Type: EBS
- Device: xvdf
- Size: 500 GiB

6. Click **Next: Add Tags**.

Step 5: Add Tags displays. This step is where you configure tags that will allow you to label and better manage your resources. For the configuration of the self-hosted version, we will not add any tags.

7. Click **Next: Configure Security Group**.

Step 6: Configure Security Group displays. This step is where you can configure the security group that will control traffic in and out of your EC2 instance. If users can access your EC2 instance from the public internet, it is very important that you use security groups to manage access.

8. Select one of the configuration methods:

To create a new security group:

- a. Select the **Create a new security group** radio button.
- b. Click **Add Rule** and create each of the following rules:

| Rule # | Purpose | Type | Protocol | Port Range | Source |
|--------|---|-----------------|----------|------------|-----------------|
| 1 | Access self-hosted from a remote desktop | RDP | TCP | 3389 | Your IP address |
| 2 | Access self-hosted version from a browser | HTTP | TCP | 80 | 0.0.0.0/0 |
| 3 | Access self-hosted version from a browser | HTTPS | TCP | 443 | 0.0.0.0/0 |
| 4 | Required for Web installer to run in on this port in HTTP | Custom TCP Rule | TCP | 8080 | 0.0.0.0/0 |
| 5 | Required for Web installer to run on this port in HTTPS | Custom TCP Rule | TCP | 8443 | 0.0.0.0/0 |

Note: Your security configuration may include the RDP rule by default; if that is true, make sure to add your IP address in the Source text field.

Note: For all the rules, keep the default drop-down menu setting of **Custom** in the Source column.

This screenshot shows what the page should display if you add the recommended rules:

| Type ⓘ | Protocol ⓘ | Port Range ⓘ | Source ⓘ |
|----------------|------------|--------------|--|
| RDP ▾ | TCP | 3389 | Custom ▾ <input type="text" value=""/> |
| HTTP ▾ | TCP | 80 | Custom ▾ 0.0.0.0, ::/0 |
| HTTPS ▾ | TCP | 443 | Custom ▾ 0.0.0.0, ::/0 |
| Custom TCP F ▾ | TCP | 8080 | Custom ▾ 0.0.0.0, ::/0 |
| Custom TCP F ▾ | TCP | 8443 | Custom ▾ 0.0.0.0, ::/0 |

To use an existing security group:

- a. Select the **Select an existing security group** radio button.
- b. Select the checkbox(es) next to the security groups you want to associate with your EC2 instance.

Select an existing security group

| Security Group ID | Name | Description |
|--|---------|---|
| <input type="checkbox"/> sg-3474bb7c | default | default VPC security group |
| <input checked="" type="checkbox"/> sg-078381c | | ELB created security group used when no security group is specified during ELB creation |
| <input type="checkbox"/> sg-034b5fd5 | | launch-wizard-1 |

9. Once you have configured your security group(s), click **Review and Launch**.

Step 7: Review Instance Launch is where you will finalize your EC2 configuration.

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

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

▼ AMI Details Edit AMI

AIO
AWS AIO Marketplace Submission
Root Device Type: ebs Virtualization type: hvm
If you plan to use this AMI for an application that benefits from Microsoft License Mobility, fill out the [License Mobility Form](#). Don't show me this again

▼ Instance Type Edit instance type

| Instance Type | ECUs | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance |
|---------------|------|-------|--------------|-----------------------|-------------------------|---------------------|
| | 13 | 4 | 16 | EBS only | Yes | High |

▼ Security Groups Edit security groups

Security group name launch-wizard-73

Cancel Previous Launch

10. Click **Launch**.

The Select an existing key pair or create a new key pair dialog box opens.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

I acknowledge that I have access to the selected private key file (DatadogProd.pem), and that without this file, I won't be able to log into my instance.

Cancel Launch Instances

- From the first drop-down menu, select Choose an existing key pair or create a new key pair.

To choose an existing key pair:

- Verify that **Choose an existing key pair** is selected in the top drop-down menu.
- In the Select a key pair drop-down menu, select an existing key pair.
- Select the **I acknowledge...** checkbox.

To create a new key pair:

- In the top drop-down menu, select **Create a new key pair**. The Key pair name text box displays.
- In the Key pair name text box, type the name of the key pair.
- Click **Download Key Pair**. A .PEM file will download to your desktop.
- Save the .PEM file because you will not be able to generate it again.

- Click **Launch Instances**.

The Launch Status screen displays.

Launch Status

Your instances are now launching
The following instance launches have been initiated: [View launch log](#)

Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances. Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. [Find out how to connect to your instances.](#)

▼ Here are some helpful resources to get you started

- [How to connect to your Windows instance](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: User Guide](#)
- [Amazon EC2: Microsoft Windows Guide](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

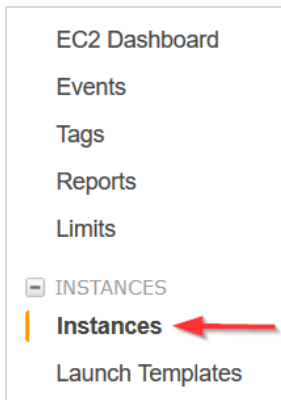
- [Create status check alarms to be notified when these instances fail status checks.](#) (Additional charges may apply)
- [Create and attach additional EBS volumes](#) (Additional charges may apply)
- [Manage security groups](#)

[View Instances](#)

This page will indicate when your EC2 instance is available. Depending on the size and scale of your deployment, it should launch within 5 to 10 minutes.

- Once the new E2 instance is generated, go back to the EC2 dashboard.

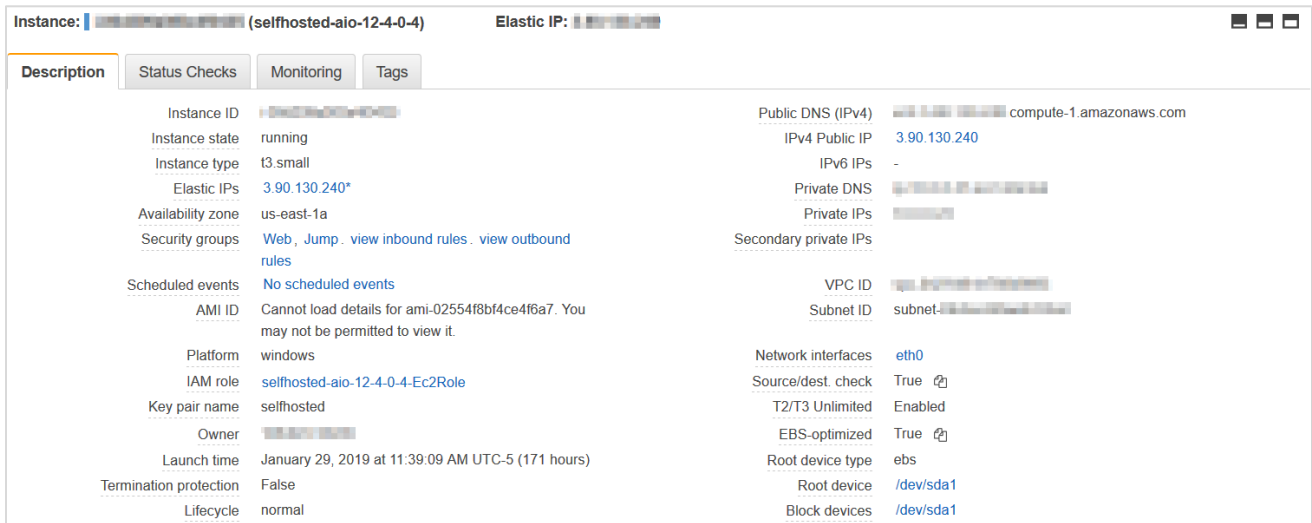
14. Select **Instances > Instances**.



The list of instances displays.

15. Select the checkbox next to your Web console EC2 instance.

The Description tab displays details about your selected EC2 instance.



16. Since these values may help you troubleshoot potential installation or performance issues, copy them to the **Required Information** section:

- Instance ID
- Instance type
- Availability zone
- Key pair (PEM file) name
- Public DNS
- Private DNS
- Subnet ID

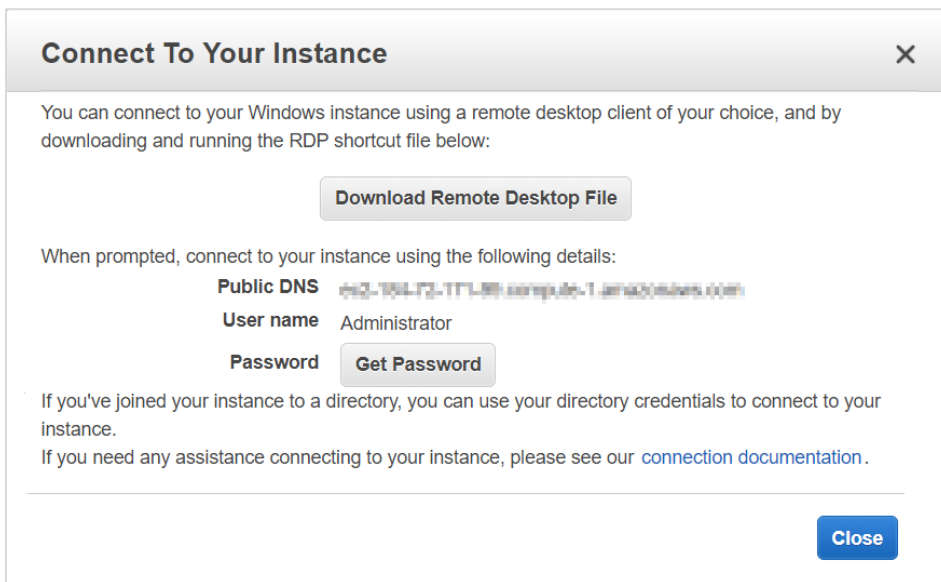
Install the Self-Hosted App

This section shows you how to install the EC2 instance.

By connecting to each EC2 instance through a Remote Desktop session, you can better manage the installation process and troubleshoot any issues that may occur.

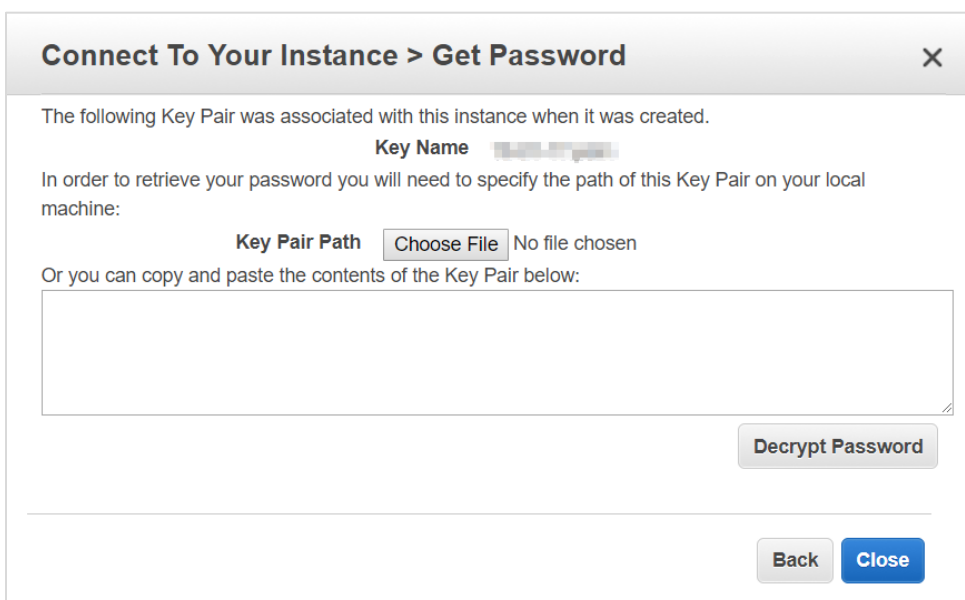
1. From your EC2 list in AWS, make sure that your Web Console EC2 instance is selected.
2. Right-click and select **Connect** from the fly-out menu.

The Connect To Your Instance dialog box opens.



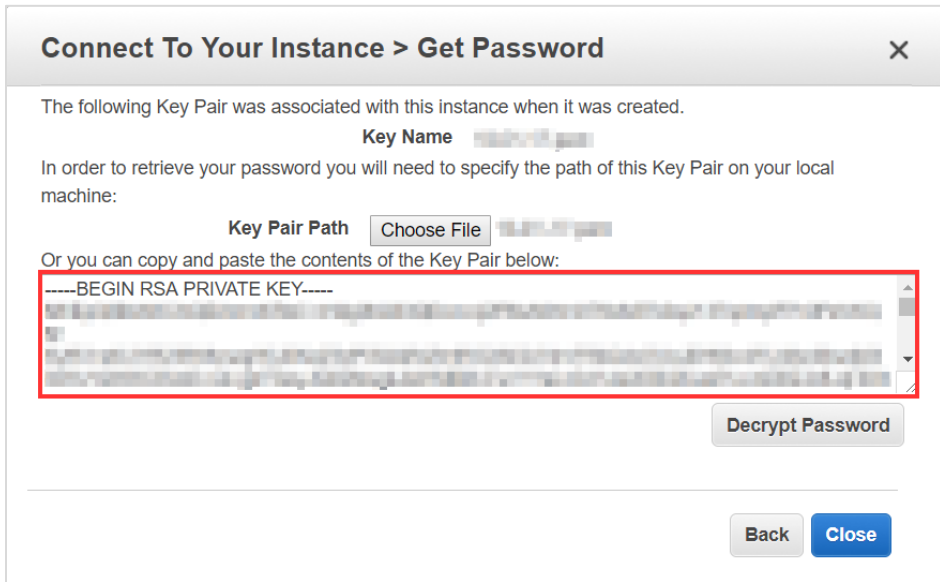
3. Click **Get Password**.


The Connect Your Instance > Get Password dialog box opens.

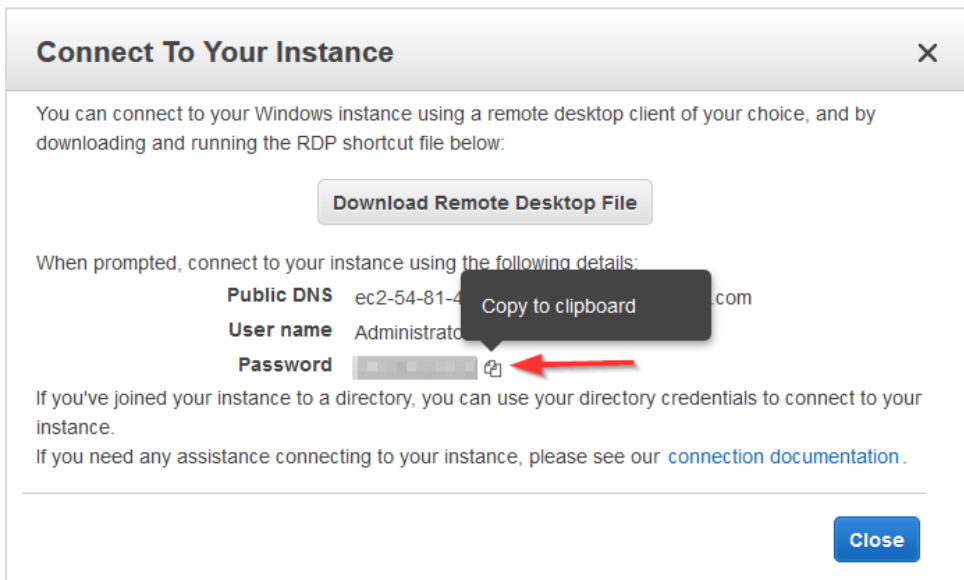


4. Click **Choose File** and navigate to location where you saved the .PEM file.
5. Click **Open**.

The contents of the file are copied over to the blank text box in the dialog box.

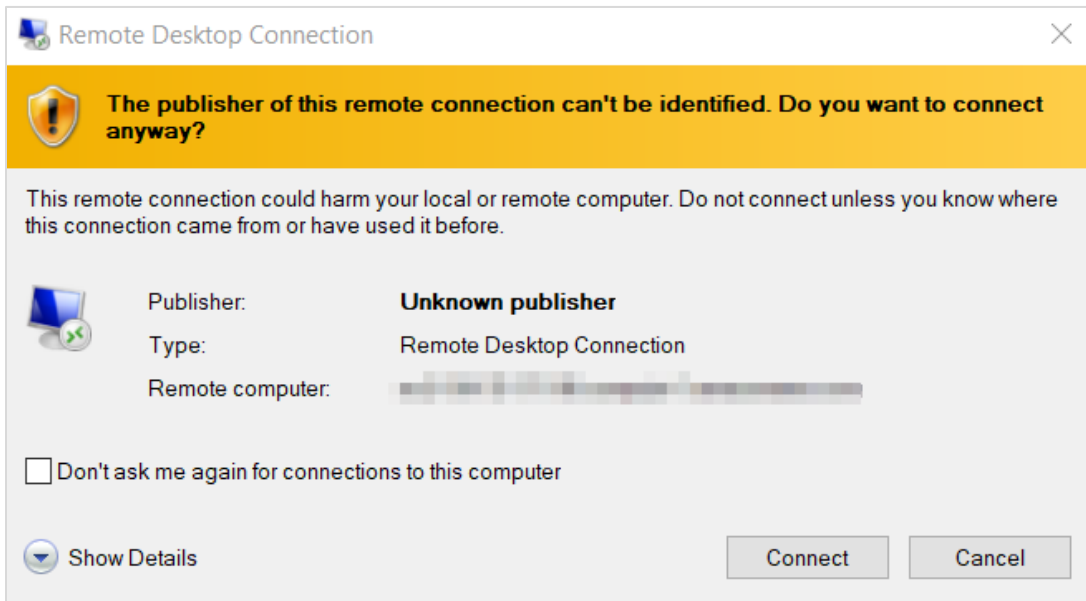


6. Click **Decrypt Password**.
- The default administrator password displays.
7. Hover to the right of the administrator password to display the **Copy to clipboard** icon.
8. Click  to save the password.



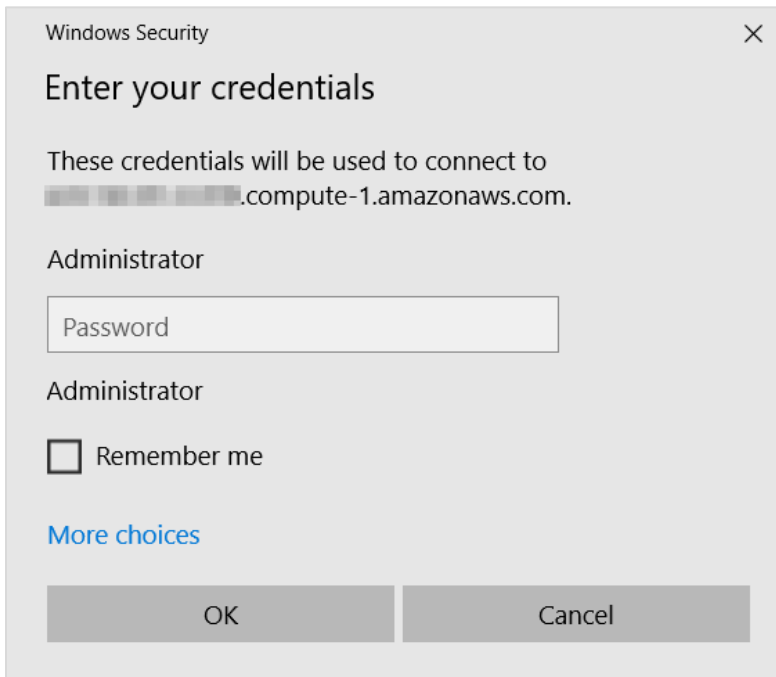
9. Click **Download Remote Desktop File**.
10. Open or save the .RDP file.

The Remote Desktop Connection dialog box opens.



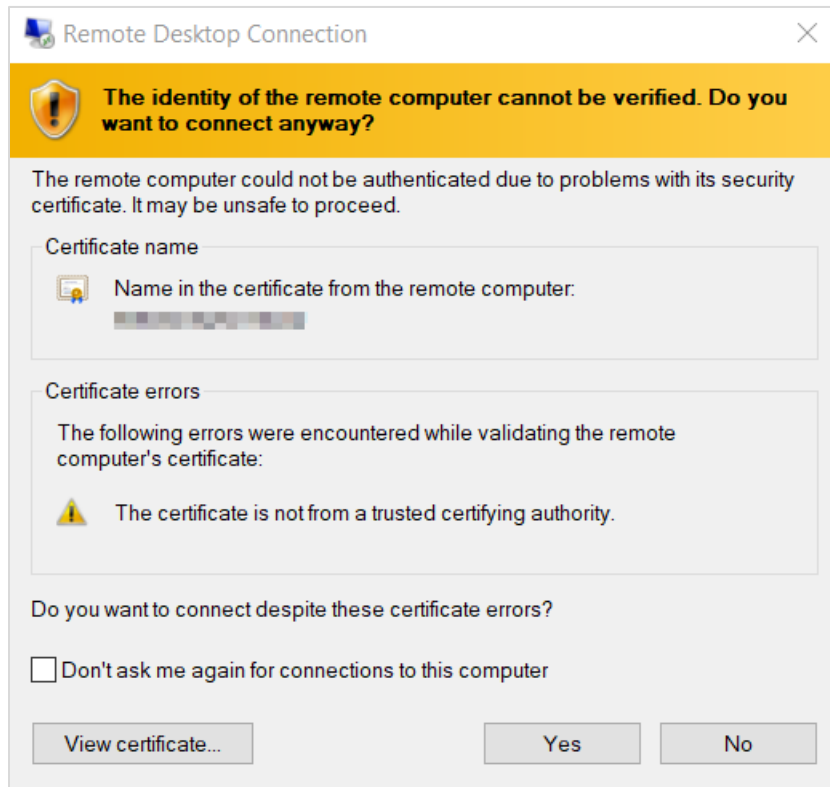
11. Click **Connect**.

The next dialog box prompts you to provide your password.



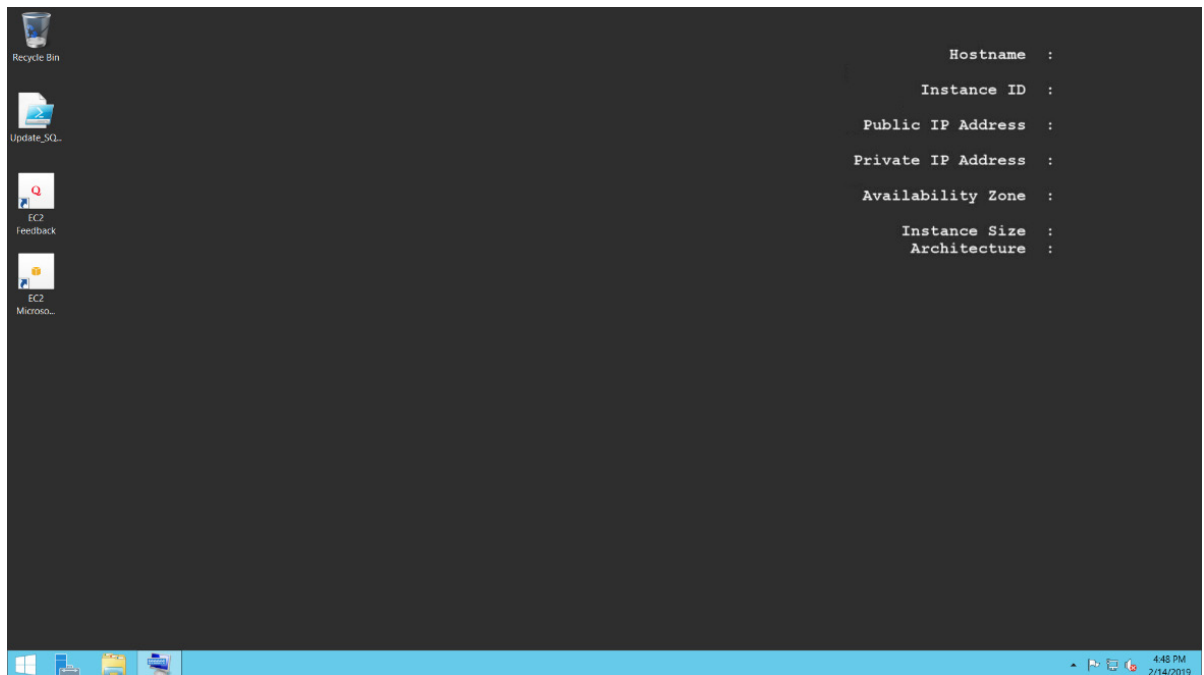
12. In the Administrator text field, paste the password you copied earlier and click **OK**.

The next dialog box prompts you to verify that you want to connect remotely.

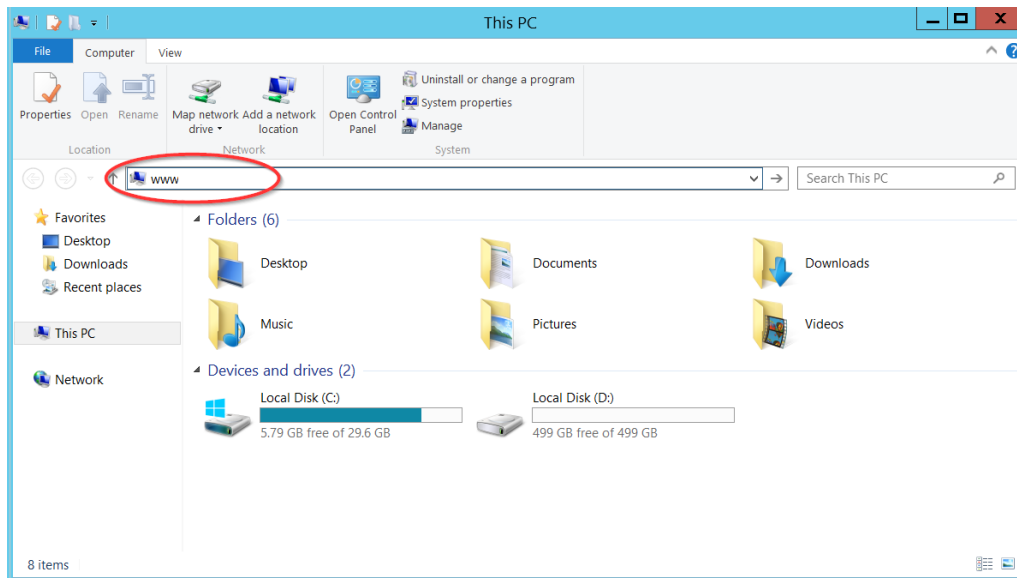


13. Click **Yes**.

Your Remote Desktop session launches.



- From the taskbar, click the **Folder** icon.
- Type **www** in the search bar to open your browser and press **Enter**.

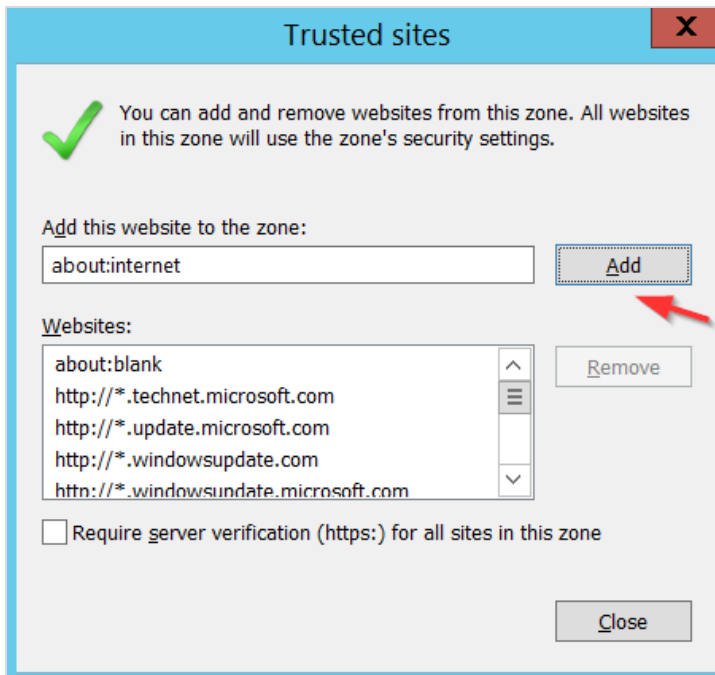


A message indicates that your browser is blocking you from reaching the internet.



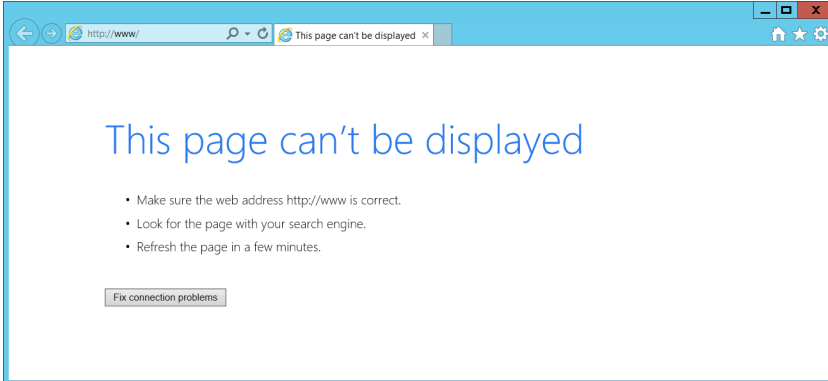
- Click **Add**. The Trusted Sites dialog box opens.

17. Click **Add** again to add this website to your list of trusted sites.



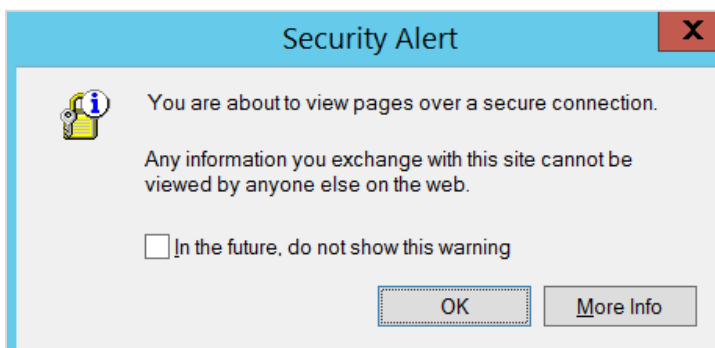
18. Click **Close**.

The browser will attempt to establish a connection with the localhost.



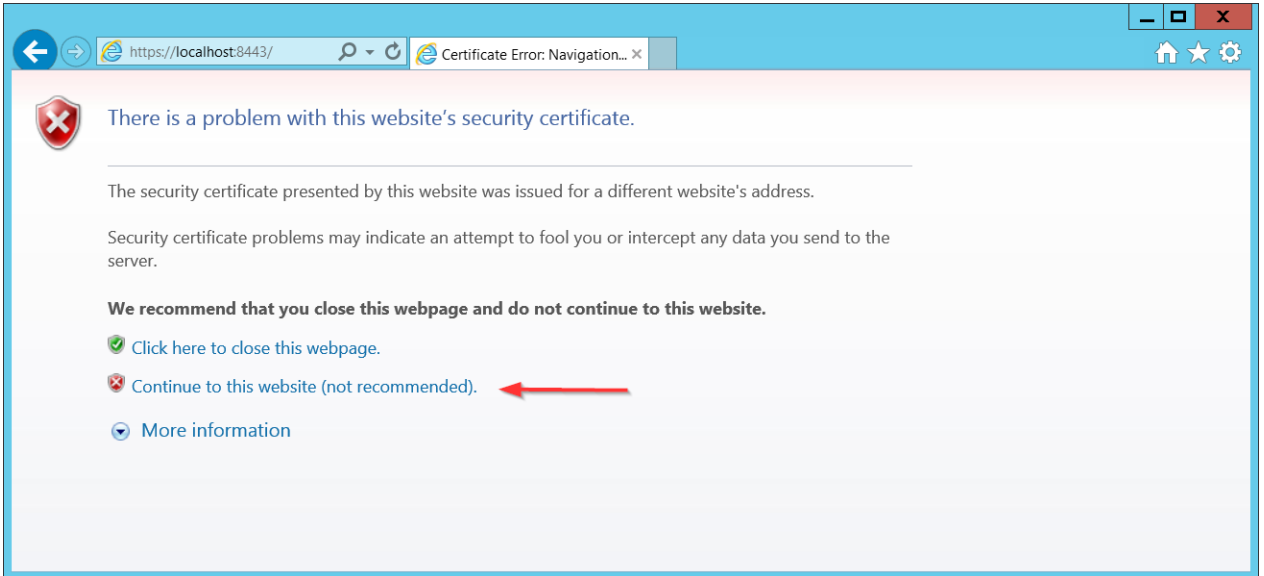
19. In the address bar, type **http://localhost:8080** and press **Enter**.

20. Click **OK** to close the security alert.



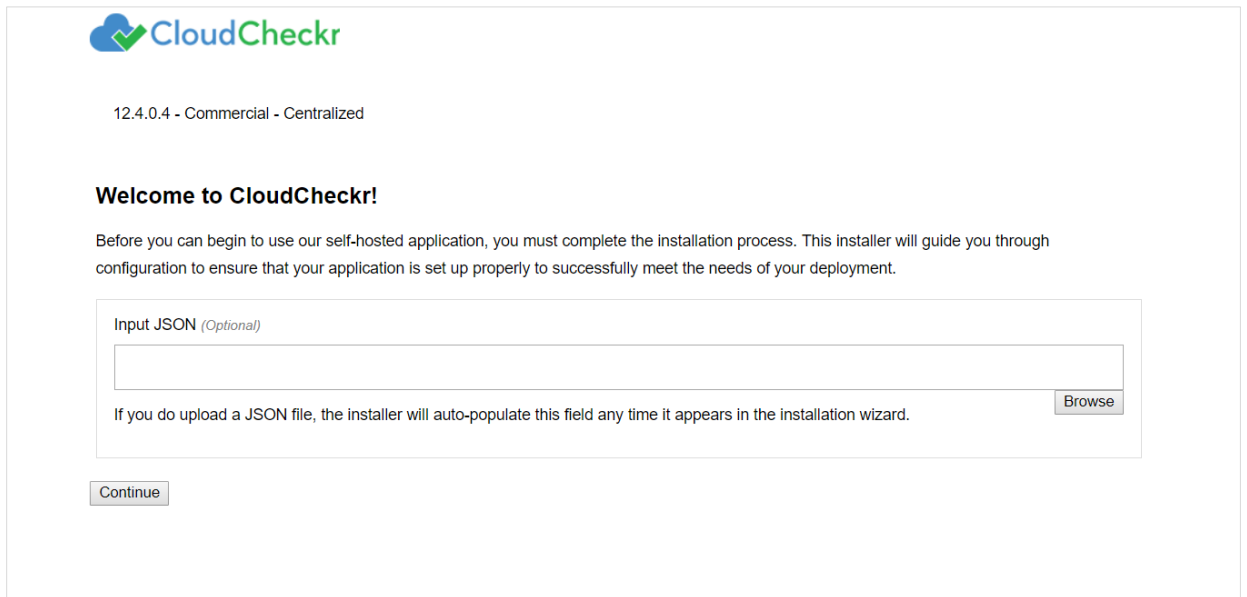
A warning about your security certificate displays.

21. Click **Continue to this website**.



22. Click **OK** to close the security alert.

The first screen of the Web installer opens. This screen is where you can upload a JSON file if you want the Web installer to auto-populate your configuration information any time it is required in the installation wizard.



Note: The Input JSON text field is an **optional** feature. If you do not want to use the website to configure the self-hosted application, you can load the file using the command line:

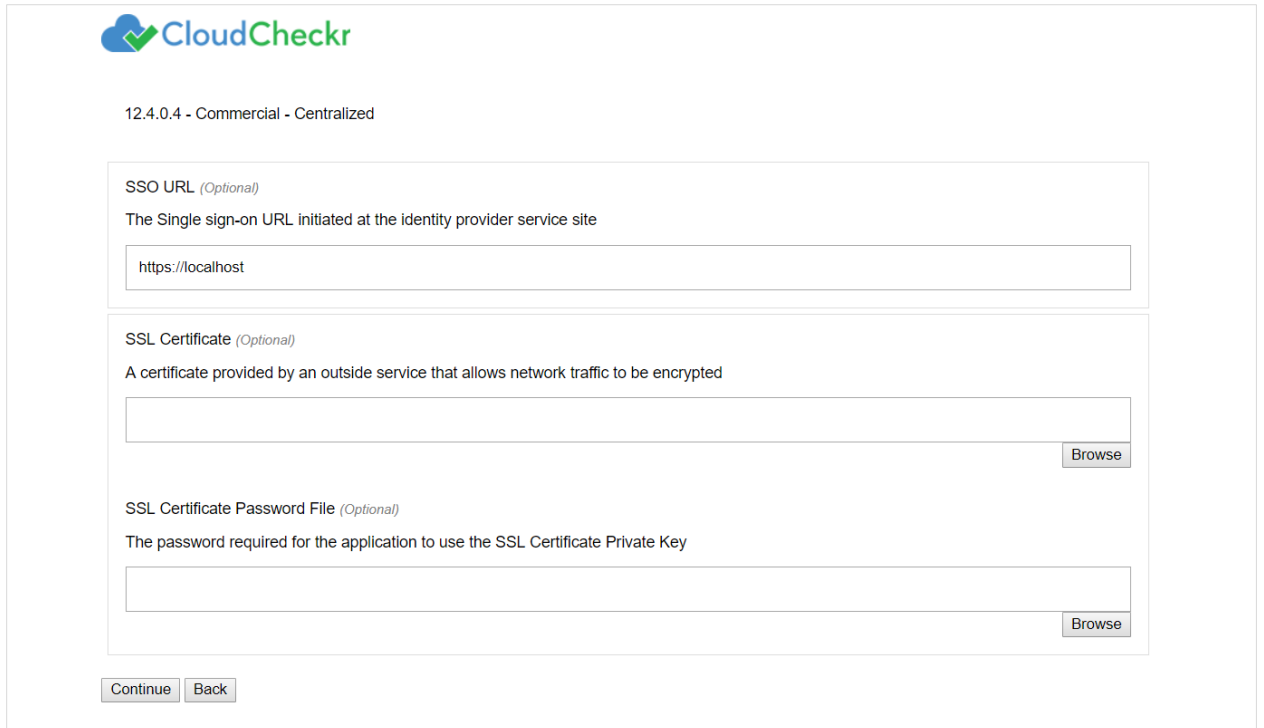
```
"C:\CloudCheckr\Package\Installer\CC.AmazonInstaller.exe - inputFile (path-to-input-file)"
```

23. If applicable, upload a JSON file by clicking **Browse** to navigate to the file location.

- See the [input JSON file section](#) for more details.

24. Click **Continue**.

The next screen in the Web installer opens. This screen is where you configure your security features.



The screenshot shows the CloudCheckr web installer interface for configuring security features. At the top left is the CloudCheckr logo. Below it, the version and deployment type are listed as "12.4.0.4 - Commercial - Centralized".

The first section is titled "SSO URL (Optional)" and includes the instruction "The Single sign-on URL initiated at the identity provider service site". A text input field contains the value "https://localhost".

The second section is titled "SSL Certificate (Optional)" and includes the instruction "A certificate provided by an outside service that allows network traffic to be encrypted". It features a large empty text input field and a "Browse" button to the right.

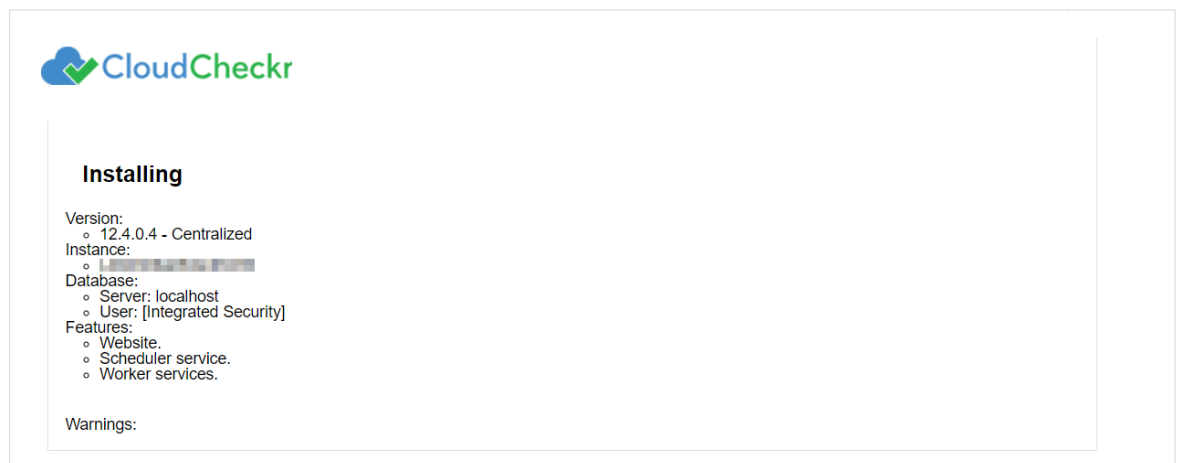
The third section is titled "SSL Certificate Password File (Optional)" and includes the instruction "The password required for the application to use the SSL Certificate Private Key". It features a large empty text input field and a "Browse" button to the right.

At the bottom of the form are two buttons: "Continue" and "Back".

25. Click **Continue**.

The next screen in the Web installer opens. The first section in this screen:

- identifies the version number of the self-hosted application
- provides the EC2 Instance ID
- verifies that the Microsoft SQL[®] server is available to communicate with the application
- identifies that the website (Web Console) is being installed



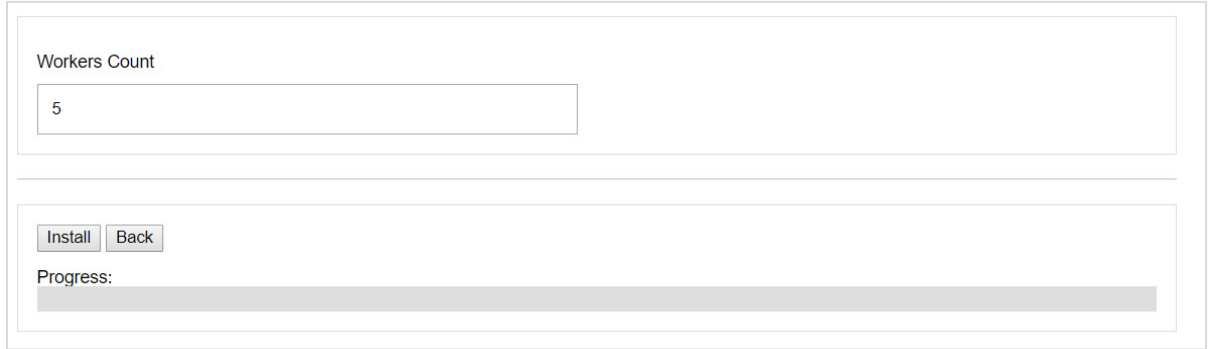
The screenshot shows the "Installing" screen of the CloudCheckr web installer. At the top left is the CloudCheckr logo. The main heading is "Installing".

The screen displays the following configuration details:

- Version:
 - 12.4.0.4 - Centralized
- Instance:
 - [REDACTED]
- Database:
 - Server: localhost
 - User: [Integrated Security]
- Features:
 - Website.
 - Scheduler service.
 - Worker services.

At the bottom, there is a section labeled "Warnings:" which is currently empty.

The second part of the screen identifies the default number of **workers**, which are Microsoft Windows® services that collect your AWS data via the API.



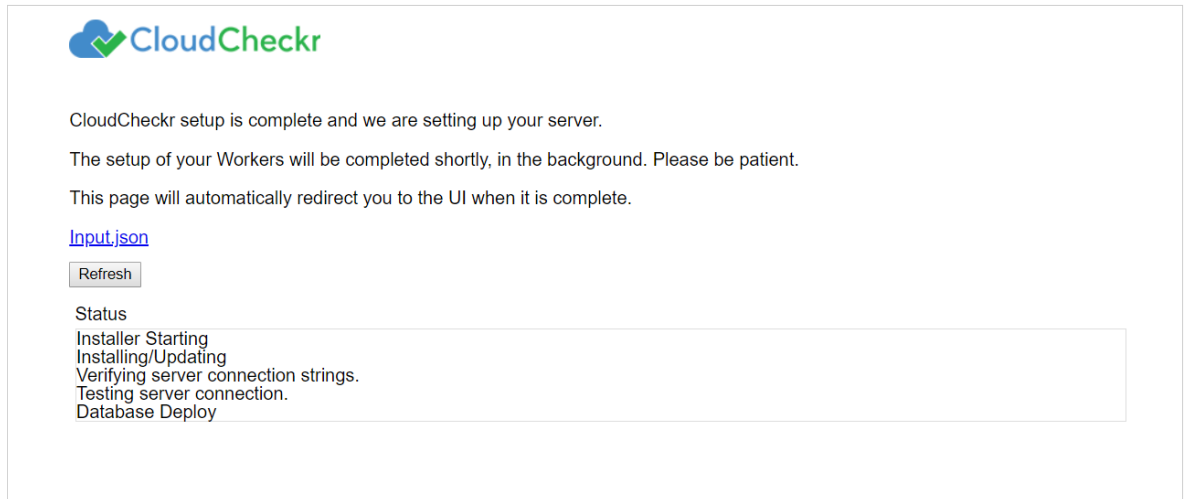
Workers Count


Progress:

26. Leave the default worker count at **5** and click **Install**.

The next screen of the Web installer updates automatically as the following tasks are completed:

- Setup and testing of server settings
- Setup of database where user accounts and data get stored
- Installation of the console (Web and application UI)
- Configuration of the workers:



 CloudCheckr

CloudCheckr setup is complete and we are setting up your server.

The setup of your Workers will be completed shortly, in the background. Please be patient.

This page will automatically redirect you to the UI when it is complete.

[Input.json](#)

Status

- Installer Starting
- Installing/Updating
- Verifying server connection strings.
- Testing server connection.
- Database Deploy

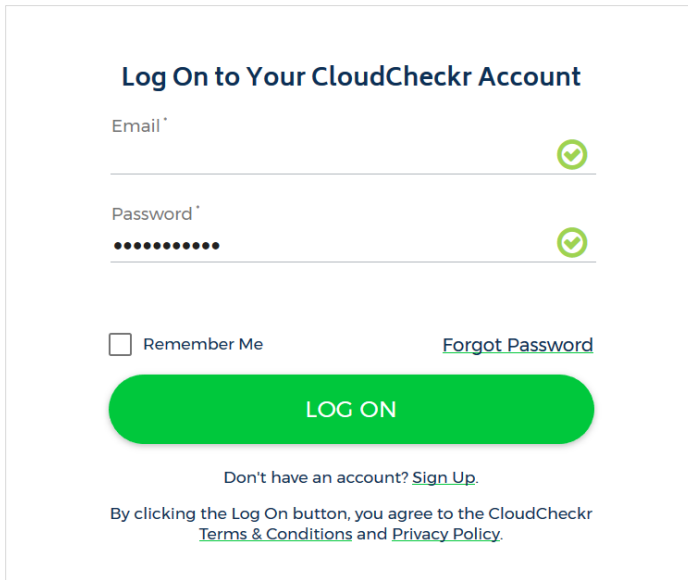
At this point, you can download the Input JSON file for later use.

The file contains the exact configuration that you set up earlier in the installation process. Since the filename is not important as part of ingestion, feel free to rename the file. If you forget to click the **Input.json** link, and you want to use the file later, you can find it on the machine at:

C:\CloudCheckr\Input.JSON

Note: The installation process may take a few minutes because the application must install the Microsoft Windows® services as well as deploy and populate the databases with the appropriate data.

The log in screen of the application opens.



Log On to Your CloudCheckr Account

Email ✓

Password ✓

Remember Me [Forgot Password](#)

LOG ON

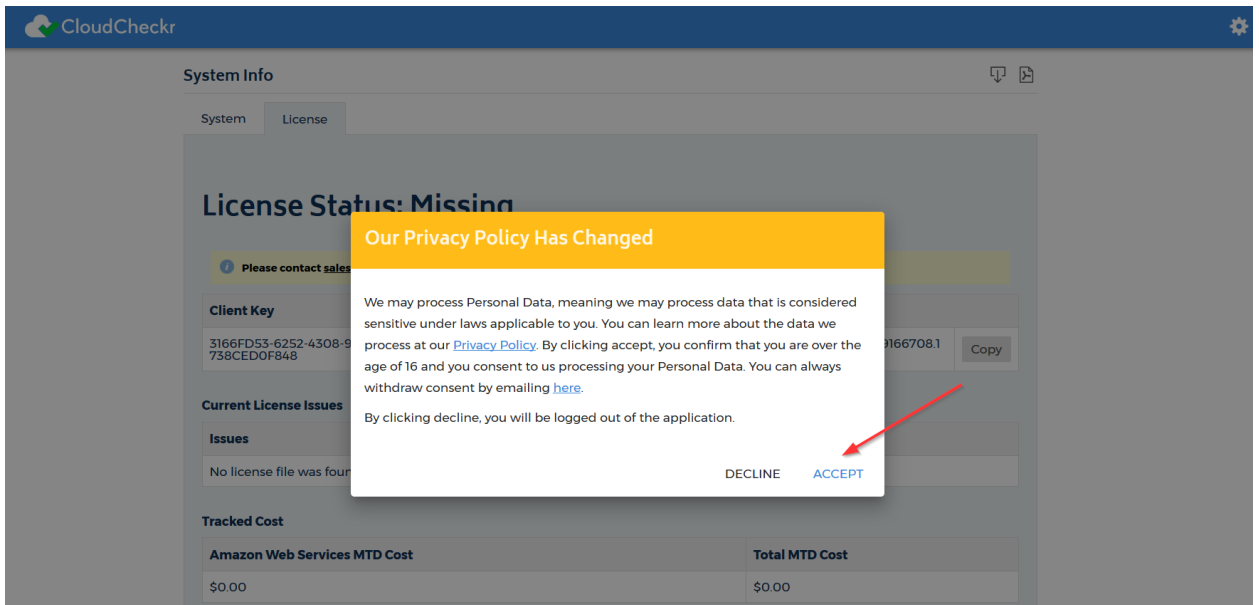
Don't have an account? [Sign Up](#).

By clicking the Log On button, you agree to the CloudCheckr [Terms & Conditions](#) and [Privacy Policy](#).

27. In the Email text field, type **sysuser**
28. In the Password text field, paste the **EC2 instance ID** of the Web console.
29. Click **LOG ON**.

When you install the AMI product for the first time, you will see our privacy notice in the foreground. The System Info page is displayed in the background.

30. Click **Accept** to acknowledge the changes to our privacy policy.



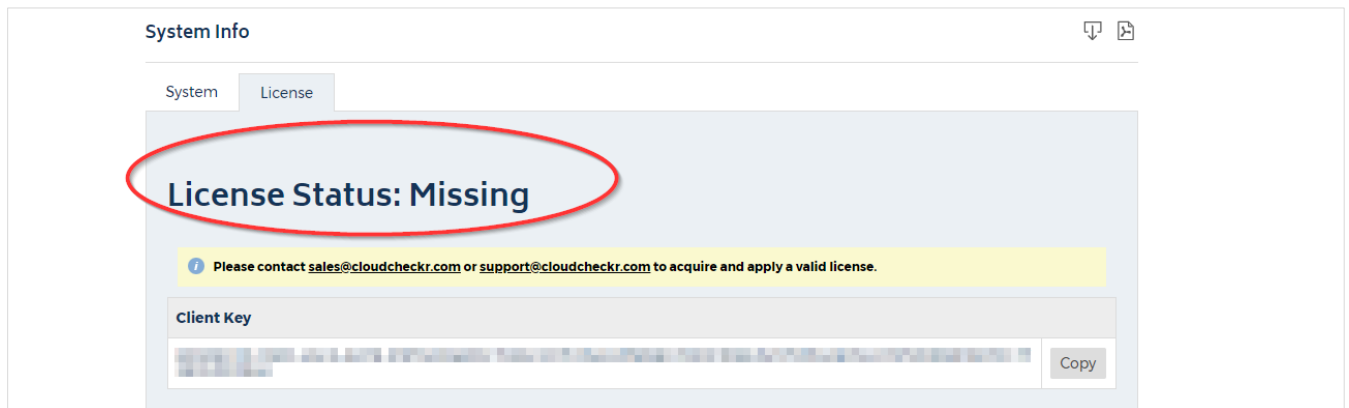
Configure the Self-Hosted App

License Your App

With the launch of the 12.4 release of the AMI and moving forward, all private offerings of the AMI product will require a license.

This section describes how to license your self-hosted application for a new installation. For more information on how to license an existing installation, contact your sales representative.

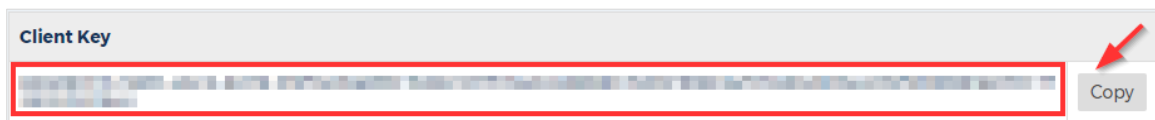
After you acknowledge the privacy policy changes, you will see is the System Info page with the License tab displayed by default. The message in the License tab will indicate that the license status is **Missing**:



You will not be able to finish the configuration of the self-hosted application until you contact sales for a new license file.

Note: To minimize the support team’s workload and reduce the steps in the license process, we prefer that customers contact their assigned sales representative first. If you are not assigned to a sales representative or you cannot reach them, email sales directly at sales@cloudcheckr.com.

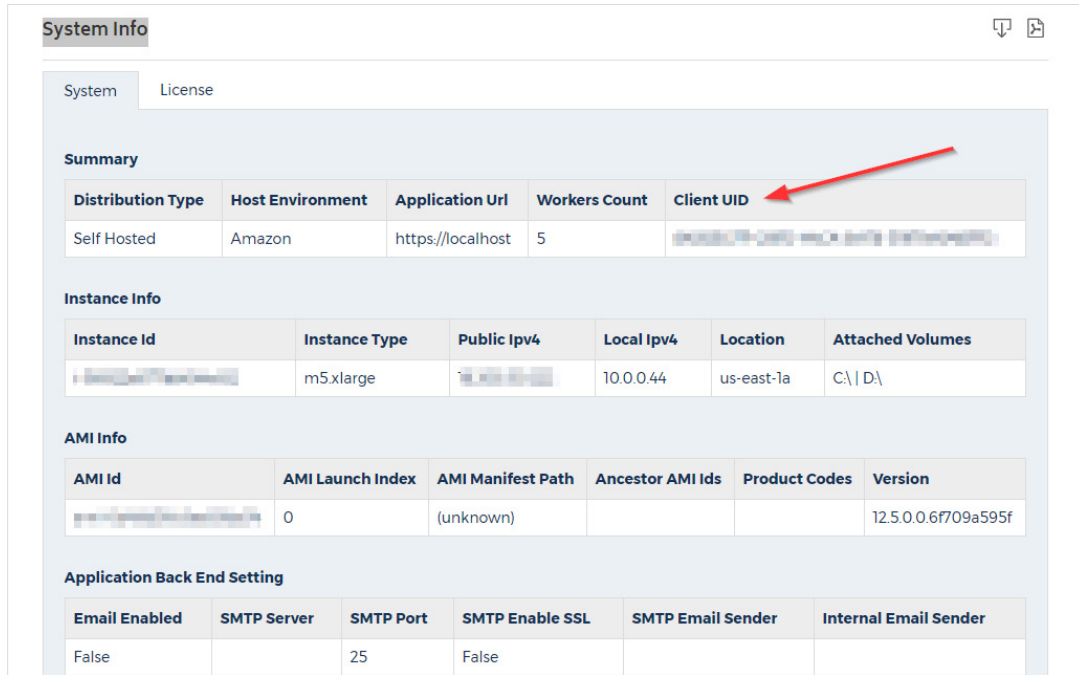
1. In the Client Key section, click **Copy** and paste the value into the Required Information table.



2. Email your sales representative to request a new license. Make sure to include the **client key** in your email.

If your organization does not allow email for security reasons, you can provide the Client UID over the phone.

- a. Locate the Client UID on the System tab. This is a unique ID or user ID that is associated with your self-hosted EC2 instance. Since it is a short string, it is easier to provide verbally than the client key, which is much longer.

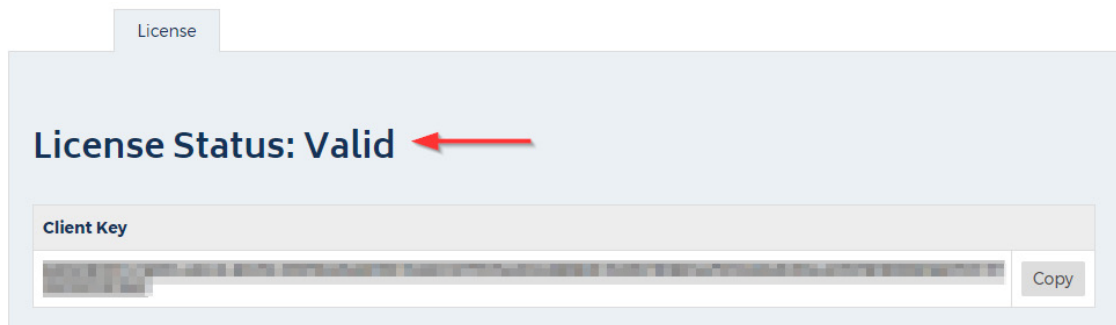


The screenshot shows the 'System Info' page with the 'System' tab selected. The 'Summary' section contains a table with the following data:

| Distribution Type | Host Environment | Application Url | Workers Count | Client UID |
|-------------------|------------------|-------------------|---------------|------------|
| Self Hosted | Amazon | https://localhost | 5 | [Redacted] |

A red arrow points to the 'Client UID' column. Below this are sections for 'Instance Info', 'AMI Info', and 'Application Back End Setting', each with their respective tables.

- b. Call your sales representative and provide the Client UID string.
3. Once your sales representative provides you with a new license file, upload the license file:
 - a. Remote into your self-hosted application.
 - b. Make sure to save the license file to a location on this machine. If you save the file to your local desktop, you will not be able to access it during your remote desktop session.
 - c. Make sure you save the file with the **LIC** extension and not as a **TXT** file. The upload will fail if the file is not saved with the correct extension.
 - d. Navigate to the License tab on the System Info page.
 - e. In the Update License section, click **Browse** and navigate to the location where you saved the license file. Once the application loads the license file, the License Status changes to **Valid**.



The screenshot shows the 'License' tab with the text 'License Status: Valid' in large blue font. A red arrow points to the word 'Valid'. Below this is a 'Client Key' section with a long redacted string and a 'Copy' button.

Create a Partner

Once your license is valid, you need to create a **partner**, which is the top-level container in the self-hosted application where you will store your accounts. For most self-hosted configurations, you will only need one partner—especially if you want to have all your accounts in one location.

1. On the License tab, click the **Back to Partners List** button.

The screenshot shows the 'License' tab in the CloudCheckr interface. At the top, it says 'License Status: Valid'. Below this, there is a 'Client Key' section with a text field containing a long alphanumeric string and a 'Copy' button. The 'Current License Info' section contains a table with the following data:

| Client UID | Created | Plan | Expiration | Max Spend / Month | Spend Warning Days |
|------------|----------|------|------------|-------------------|--------------------|
| [REDACTED] | 3/1/2019 | test | 4/30/2019 | \$50,000.00 | 10 |

The 'Tracked Cost' section shows a table with two columns: 'Amazon Web Services MTD Cost' and 'Total MTD Cost', both showing '\$0.00'. Below this is an 'Update License' section with instructions to upload a license file. A 'License File' section has a 'Browse' button and the text '[none selected]'. At the bottom left is a green 'Save' button, and at the bottom center is a blue 'Back To Partners List' button, which is highlighted by a red arrow.

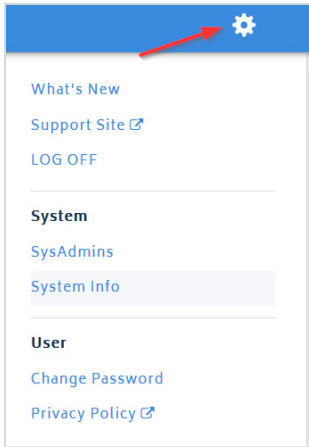
The Partners Landing page opens.

The screenshot shows the 'Partners' landing page in the CloudCheckr interface. At the top, there is a 'Partners' section with a '+ NEW PARTNER' button and a search filter. The search filter has fields for 'Id', 'Name', and 'Email', and a checkbox for 'Include children'. Below the search filter is a green 'Filter' button. The main content is a table with the following data:

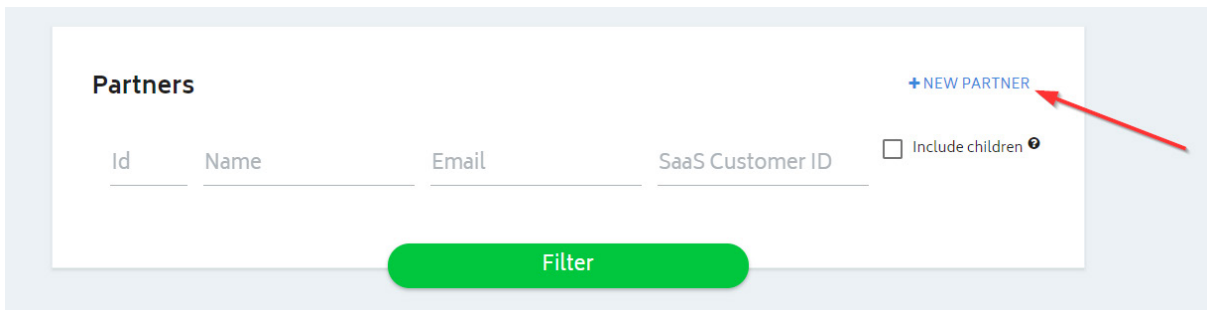
| ID | PARTNER NAME | EMAIL | CURRENT PLAN | USERS | SIGNUP TYPE | NOTES | AUDIT HISTORY | ACTIONS |
|----|--------------|-------|--------------|-------|-------------|-------|---------------|---------------|
| 1 | Systemjobs | | Paid | 1 | Standard | Edit | View | [Edit] [View] |

At the bottom left, it says 'Showing 1 - 1 of 1'. At the bottom right, there is a pagination control showing '1' and a 'Show: 50' dropdown.

Only the Settings icon is in the header bar—indicating that you have not fully activated the self-hosted application. If you were to click the **Settings** icon, you could only perform basic tasks and view your system and license information.

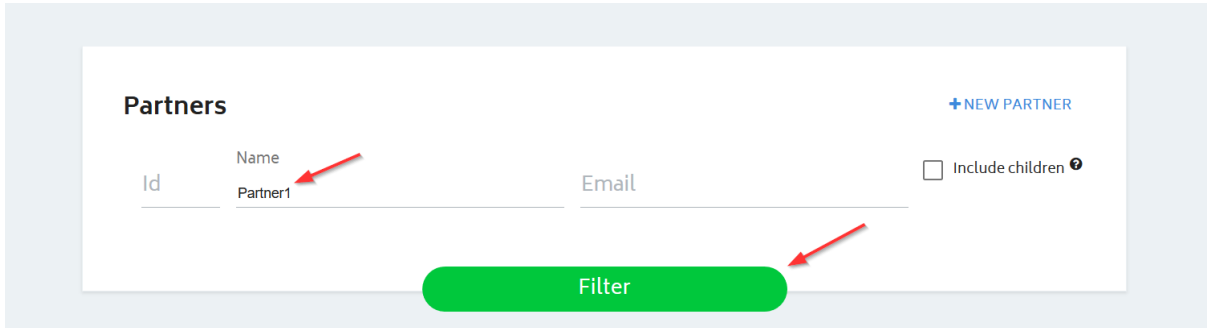


2. Click **+ NEW PARTNER**.



The Add Partner dialog box opens.

- In the Partner Name text field, type a partner name.
 - In the Partner Email text field, type an email address.
 - Click **Create**.
- A message indicates that the application added the partner successfully.
- Click **OK**.
 - Type the name of your new partner and click **Filter**.

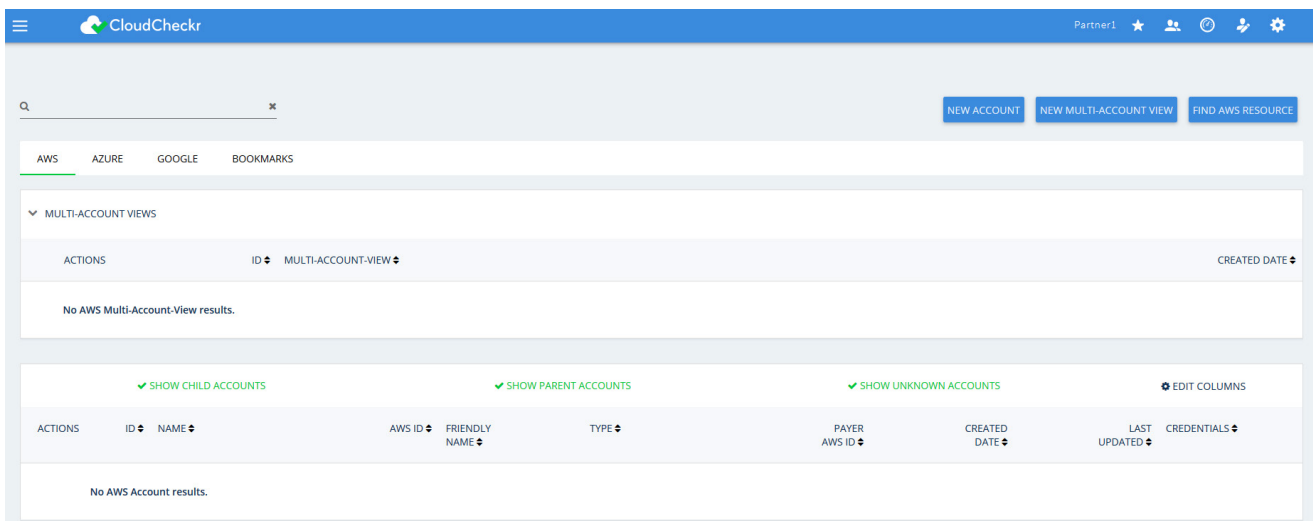


Your new partner is now included in the partner list.





| ID | PARTNER NAME | EMAIL | CURRENT PLAN | USERS | SIGNUP TYPE | NOTES | AUDIT HISTORY | ACTIONS |
|----|--------------|-----------------------------|--------------|-------|-------------|-------|---------------|---|
| 1 | SystemJobs | | Paid | 1 | Standard | Edit | View | Edit Delete |
| 2 | Partner1 | mary.norman@cloudcheckr.com | Paid | 0 | Standard | Edit | View | Edit Delete |

Even though you have created your partner, you still only have basic functionality. If you drill into the partner, you will have access to more features.

- Click the **partner name**.
- Now you are on the Accounts page.



Notice that more icons are now available in the header bar. Here's a quick overview:

| New | Description |
|---|---|
|  | Create or access bookmarks to application features. |
|  | Return to the Partners landing page. |
|  | Create or access custom dashboards. |
|  | Modify or view application settings. |

Before you can create your account(s) for this partner, you need to finish some back-end configuration steps.

Complete the Back-End System Configuration

In its default state, the self-hosted application does not have the same functionality as the SaaS version.

Follow these instructions to complete the application configuration.

1. From the menu bar, click the **Settings** icon and choose **System > Configuration**.

The Application-wide Configurations page opens.

Application-wide Configurations

These configuration settings apply to all workers, web portal, and databases running for this CloudCheckr application.

SMTP Settings

In order for CloudCheckr to send emails, you will need to configure the SMTP server, username, password, and other details that CloudCheckr will use to send the outbound emails.

Email Enabled

SMTP Server

SMTP Port

SMTP User

SMTP Password Is set: ✗

SMTP Enable SSL

SMTP Email Sender

Internal Email Sender

URL For CloudCheckr

CloudCheckr generates URLs that may be delivered in emails or in various sections of the web portal.

In order for these links to work, you will need to set the initial part of the URL to let CloudCheckr know what hostname to use.

Use the URL:

2. Scroll down to the SMTP section and configure the settings that will allow the self-hosted application to send emails on new user activations, alerts, and report data.
3. Scroll down to the URL For CloudCheckr section and provide the URL that you want to display on any system-generated emails.

Note: The default **localhost** will display the DNS for the EC2 instance that is hosting your self-application. This URL is external-facing so you can use it to send emails.

4. Scroll down to the Workers section to see the default number of workers that the self-hosted application will install.

Note: The default number of workers in a self-hosted environment is **5**. If you increase the number of AWS accounts in your self-hosted environment to 10-30, you can increase the number of workers to 25.

Workers

NOTE: Any change in workes will stop all Cloudcheckr services and start CloudCheckr_Installer service in the instance.

Instanceld: i-00d309a083e4f0435 Scheduler Workers Count:

5. Scroll down to the Contact Info for CloudCheckr section and change the default email addresses and phone number if you want your users to contact you directly.

Contact Info For CloudCheckr

CloudCheckr displays warn messages and help text from time to time, with our Email and Phone Number. If this contact info needs to be updated so your users can contact you directly, you can edit that contact info here.

Sales Email Address:

Support Email Address:

Development Email Address:

Phone Number:

6. Scroll down to the Proxy section to enable your proxy configuration settings.

Proxy

If you are running CloudCheckr on a network that requires proxy configuration to reach the AWS API, you can enable those settings here.

Proxy Credentials Domain

Proxy Credentials UserName

Proxy Credentials Password

Proxy Host

Proxy Port

Ignore Certificate Validation when proxying connections

7. Scroll down to the Credentials for Updating AWS Prices section to paste the values of the access and secret keys you created in the Create IAM Users section.

Credentials for Updating AWS Prices

In order for CloudCheckr to stay up-to-date with the AWS pricing, CloudCheckr needs to connect to the AWS API and pull down the latest pricing. CloudCheckr will need credentials to do that.

The credentials you enter should have access to:

ec2:DescribeAvailabilityZones
ec2:DescribeReservedInstancesOfferings

Credential 1

AWS Account:

Access Key ID

Secret Access Key

Credentials are for a GovCloud account

Credential 2

AWS Account:

Access Key ID

Secret Access Key

Credentials are for a GovCloud account

Credential 3

AWS Account:

Access Key ID

Secret Access Key

Credentials are for a GovCloud account

Create a Trusted User

1. Scroll down to the AssumeRole section on the Application-wide Configurations page.

When you assume a role, AWS gives you temporary security credentials to access other AWS accounts. The name of this role is referred to as a **cross-account role**.

Before you can create a cross-account role, you must plug in the access and secret keys of a Trusted User. A **Trusted User** is an IAM user whose credentials will enable the cross-account role to work with the self-hosted application.

AssumeRole

Enter the default AWS Credentials that will be used to assume role in your accounts.

IMPORTANT! If this credentials are to assume role in a **Custom Region**, make sure you first set and save that region.

Credential

AWS Account:

Access Key ID

Secret Access Key

2. To create a trusted user:
 - a. Copy the Trusted User policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1474398174000",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::AWS ACCOUNT ID:user/root"
      ]
    }
  ]
}
```

- b. Replace **AWS ACCOUNT ID** with the 12-digit AWS account ID associated with your AWS account.
 - c. Create and save the Trusted User policy using the instructions in the [Create an IAM Policy](#) section.
 - d. Create a Trusted User group and attach the Trusted User policy using the instructions in the [Create an IAM Group](#) section.
 - e. Create a trusted user and attach it to the Trusted User group using the instructions in the [Create an IAM User](#) section.
 - f. Paste the values of the access and secret keys of the Trusted User into the AssumeRole section.
 3. Click **Save Settings** to save all the configuration changes you made to the self-hosted application.
 4. Copy the name of the Trusted User Policy, Trusted User Group, and Trusted User to the [Required Information](#) section.

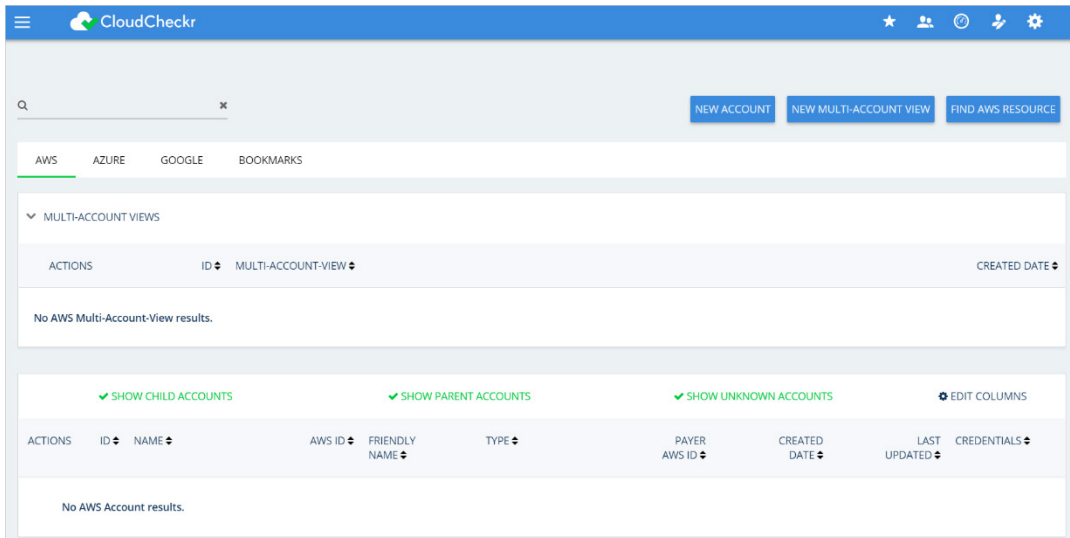
Create an Account

Now that you have configured all the back-end settings for the self-hosted application, you need to create an account or accounts within that partner. The account is where you will perform all your work in the self-hosted application—such as running reports, configuring alerts, and creating invoices.

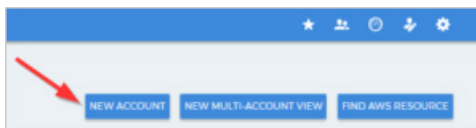
1. From the Application-wide Configurations page, click **Back to Accounts**.

Note: You can also go to the Partners landing page and click **your partner name**.

The Accounts page for your partner displays.



2. From the right side of the screen, click **NEW ACCOUNT**.



The New Account screen displays.

New Account

Enter a name for your Account: 0 / 256

Cloud Provider
Select the cloud services provider:
Amazon Web Services ▾

Navigation Visibility
Select the sections you want users to see in this account. You can change these settings at any time.

- Recently Viewed
- Savings
- Best Practices
- Cost
- Inventory
- Security
- Utilization
- Automation

Create **Cancel**

3. Type a unique name for your account and in the Cloud Provider section, select **Amazon Web Services**.
4. Scroll down to the Navigation Visibility section, and select the modules that you want your account to have access to:
 - **Recently Viewed:** shows the 10 reports that were most recently accessed
 - **Savings:** shows you how to save the most amount of money in the shortest amount of time
 - **Best Practices:** contains more than 550 recommendations based off the industry compliance standards
 - **Cost:** includes all reports on your daily spend, Reserved Instances (RIs), access to raw billing data, and more
 - **Inventory:** contains list of Summary, Detail, and Trending reports on your cloud provider's offerings
 - **Security:** helps you audit, conduct forensics, and manage other security issues for your cloud deployment
 - **Utilization:** provides metrics, visualization, analysis, and right-sizing recommendations for your environment
 - **Automation:** provides functionality to automate administrative tasks related to security and maintenance
5. At the bottom of the New Account page, click **Create**.

The Configure Account page opens.

By default, the Use a Role for Cross-Account Access tab is selected. This tab is visible because you provided the access key and secret key for the Trusted User in the [Create a Trusted User](#) procedure.

Configure Account Show Help ☆

Use a Role for Cross-Account Access Use an IAM Access Key Map To Payer

Select the AWS Account type below:

Credentials are for a Standard (Commercial) account

[Toggle Manual vs. CloudFormation](#)

1. In the Billing & Cost Management Dashboard of the AWS Management Console, verify that the **Receive Billing Alerts** checkbox is selected. (optional)
2. Click the [Launch CloudFormation Stack](#) link.
3. Type a new name for your stack
4. For each of the separate policies—Inventory, Billing, Security, and CloudWatch Flow Logs—select **True** or **False** if you want to include that policy in your template.
 1. For Billing, type the name of your AWS Detailed Billing Report bucket.
 2. For Security, type the name of your AWS CloudTrail bucket.
5. Select the **I Acknowledge that AWS CloudFormation might create IAM resources** checkbox and click **Create**.
6. When the stack creation is complete, select your stack name from the list and click the **Resources** tab.
7. Click the **Physical ID** link for the IAM role.
8. From the Summary page, copy the Role ARN value.
9. Select the checkbox if this is an account from India managed by Amazon Internet Services Pvt. Ltd. ([AISPL](#)).
 This account is managed by AISPL.
10. Paste the Role ARN value in the field:
AWS Role ARN

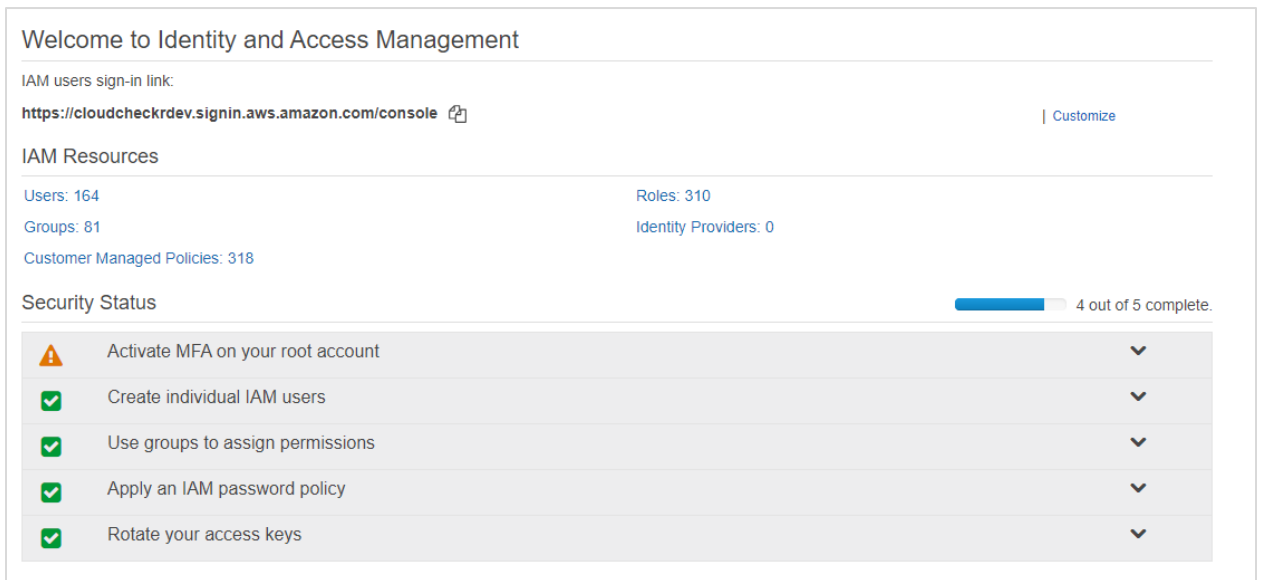
[Update](#)

To finish your account configuration, you will return to AWS to create a cross-account role.

Create an IAM Role for Cross-Account Access

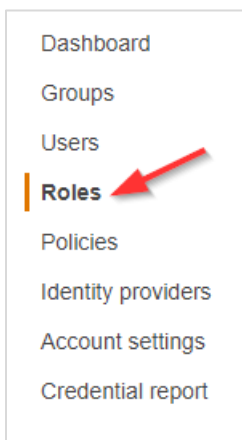
This procedure shows you how to create an IAM role in AWS and apply those credentials in the self-hosted application to complete the cross-account access configuration.

1. Log into the AWS Management Console.
The AWS services page opens.
2. Scroll down to the Security, Identity & Compliance section and select **IAM**.
The Welcome to Identity and Access Management screen displays.



The screenshot shows the AWS IAM console dashboard. At the top, it says "Welcome to Identity and Access Management". Below that, there is a link for IAM users sign-in: <https://cloudcheckrdev.signin.aws.amazon.com/console> with a "Customize" link. The "IAM Resources" section shows: Users: 164, Groups: 81, Customer Managed Policies: 318, Roles: 310, and Identity Providers: 0. The "Security Status" section shows a progress bar for "4 out of 5 complete" and a list of tasks: "Activate MFA on your root account" (warning icon), "Create individual IAM users" (checkmark), "Use groups to assign permissions" (checkmark), "Apply an IAM password policy" (checkmark), and "Rotate your access keys" (checkmark).

3. From the dashboard, click **Roles**.



The screenshot shows the navigation menu in the AWS IAM console. The menu items are: Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, and Credential report. The "Roles" item is highlighted with a vertical orange bar and a red arrow pointing to it.

The Roles page opens.

Roles

What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use Identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

Additional resources:

- IAM Roles FAQ
- IAM Roles Documentation
- Best practices for setting up cross-account access
- Tutorials on roles

Create role **Delete role**

Showing 61 results

| Role name | Description | Trusted entities |
|--|-------------|------------------|
| <input type="checkbox"/> ActiveDirectoryDB | | AWS service: rds |
| <input type="checkbox"/> ActiveDirectoryQA | | AWS service: ds |

4. From the middle of the page, click **Create role**.

The Create role page opens.

5. In the Select type of trusted entity section, click **Another AWS account**.

Select type of trusted entity

AWS service
EC2, Lambda and others

Another AWS account
Belonging to you or 3rd party

Web identity
Cognito or any OpenID provider

SAML 2.0 federation
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

The screen prompts you to add an Account ID value.

Create role

1 2 3 4

Select type of trusted entity

AWS service
EC2, Lambda and others

Another AWS account
Belonging to you or 3rd party

Web identity
Cognito or any OpenID provider

SAML 2.0 federation
Your corporate directory

Allows entities in other accounts to perform actions in this account. [Learn more](#)

Specify accounts that can use this role

Account ID*

Options

- Require external ID (Best practice when a third party will assume this role)
- Require MFA

* Required

Cancel **Next: Permissions**

- b. Copy the external ID identified in the instructions.

Use a Role for Cross-Account Access

Select the AWS Account type below:

Credentials are for a Standard (Commercial) account

Toggle Manual vs. CloudFormation

1. Log in to your AWS Management Console and access the [IAM dashboard](#).
2. Select **Policies** from the left menu and click the **Create policy** button.
3. Go to our [support site](#) and copy the policy or policies that apply to your business needs.
4. For each policy, follow these steps:
 1. Click the **JSON** tab, and paste the new policy into the tab.
 2. Click **Review Policy**.
 3. Type a name for the policy and click **Create policy**.
 4. Select the policy from the list and from the Policy actions drop-down menu, select **Attach**.

Note: For any DBR and CloudTrail policies that you create, make sure that you replace the default S3 bucket with the name of the new S3 bucket identified in the policy.

5. Select **Roles** from the left menu and click the **Create role** button.
6. Select **Another AWS account**.
7. Select the Require external ID checkbox next to Options.
8. Copy these values to the corresponding fields in AWS:

Account ID: [redacted]

External ID: [redacted]

9. Return to the AWS Management Console and perform the following steps:
 - a. Paste the external ID value that you just copied.
 - b. Verify that the Require MFA radio button is not selected.

Options Require external ID (Best practice when a third party will assume this role)

You can increase the security of your role by requiring an optional external identifier, which prevents "confused deputy" attacks. This is recommended if you do not own or have administrative access to the account that can assume this role. The external ID can include any characters that you choose. To assume this role, users must be in the trusted account and provide this exact external ID. [Learn more](#)

External ID

[External ID input field]

Important: The console does not support using an external ID with the Switch Role feature. If you select this option, entities in the trusted account must use the API, CLI, or a custom federation proxy to make cross-account iam:AssumeRole calls. [Learn more](#)

Require MFA 

10. Click **Next: Permissions**.
A list of policies displays.
11. Select the checkbox next to the policy or policies you want to attach to this role and click **Next: Tags**.
The Add tags page displays. Adding tags is **optional**. For the purposes of this procedure, we will not add tags.

12. Click **Next: Review**. The Review page opens.

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name*
Use alphanumeric and '+, @, _' characters. Maximum 64 characters.

Role description
Maximum 1000 characters. Use alphanumeric and '+, @, _' characters.

Trusted entities The account 352813966189

Policies [AAA](#)

Permissions boundary Permissions boundary is not set

No tags were added.

* Required Cancel Previous Create role

13. Type a name for the role and click **Create role**.

A message indicates that AWS has created your role.

The role is now listed with the other IAM roles associated with your AWS account.

14. Select the checkbox next to your new role and click the **role name**.

The Summary page for the selected role opens.

Roles > trusteduser

Summary

[Delete role](#)

| | |
|--|---|
| Role ARN | arn:aws:iam::123456789012:role/trusteduser |
| Role description | Edit |
| Instance Profile ARNs | copy |
| Path | / |
| Creation time | 2019-01-18 14:31 EST |
| Maximum CLI/API session duration | 1 hour Edit |
| Give this link to users who can switch roles in the console | https://console.aws.amazon.com/iam/home?#/roles/trusteduser |

At the top of the page, you will see the Role ARN value.

ARN values use this format:


```
arn:aws:iam::YourAccountIDHere:role/CloudCheckrRole
```



15. Click the **Copy** icon next to the Role ARN value.

Role ARN arn:aws:iam: [redacted]  

16. Return to Configure Accounts page in the self-hosted application and perform the following steps:
 - a. Paste the Role ARN value in the AWS Role ARN field.
 - b. Click **Update**.

15. Paste the Role ARN value in the field:

AWS Role ARN 

Update 

You now have an AWS cross-account role associated with your self-hosted application account.

17. Copy the name of the cross-account role to the [Required Information](#) section.

Create Least Privilege Policies

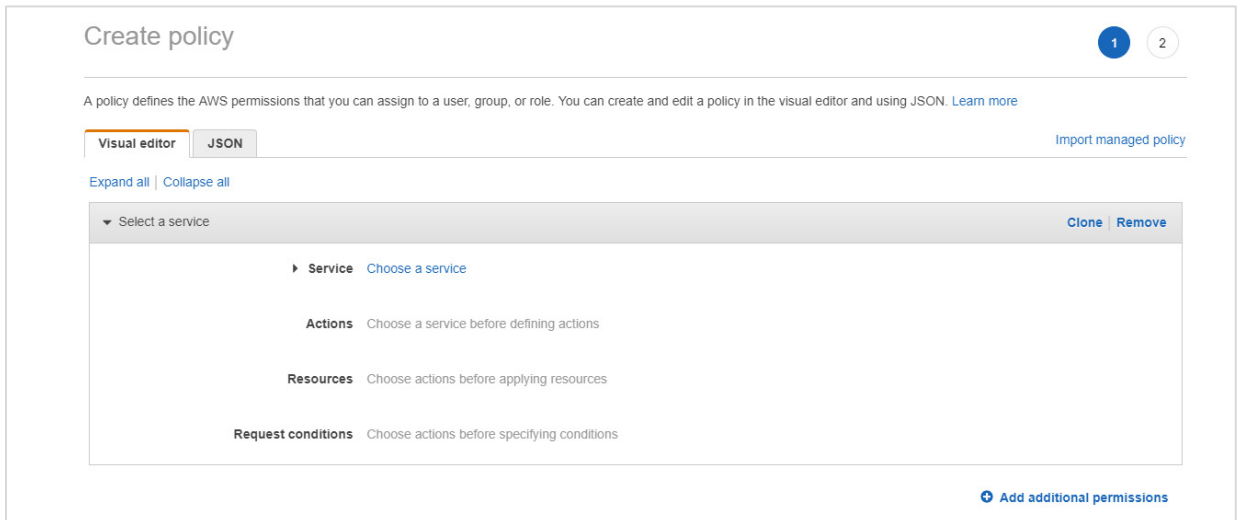
The self-hosted application must have permission to ingest the AWS data needed to populate its reports.

To provide your self-hosted application with the permissions it needs, our team has designed **least privilege policies**. Each of these policies provides permission to a core area of AWS functionality:

- Cost
- Billing
- Security/Compliance
- Inventory
- CloudWatch Flow Logs
- CloudTrail

In this procedure, you will create these least privilege policies in AWS:

1. From the AWS Services page, scroll down to the Security, Identity & Compliance section and select **IAM**.
The Welcome to Identity and Access Management screen displays.
2. From the dashboard, click **Policies**.
A list of policies displays.
3. Click **Create policy**. The Create Policy page opens.



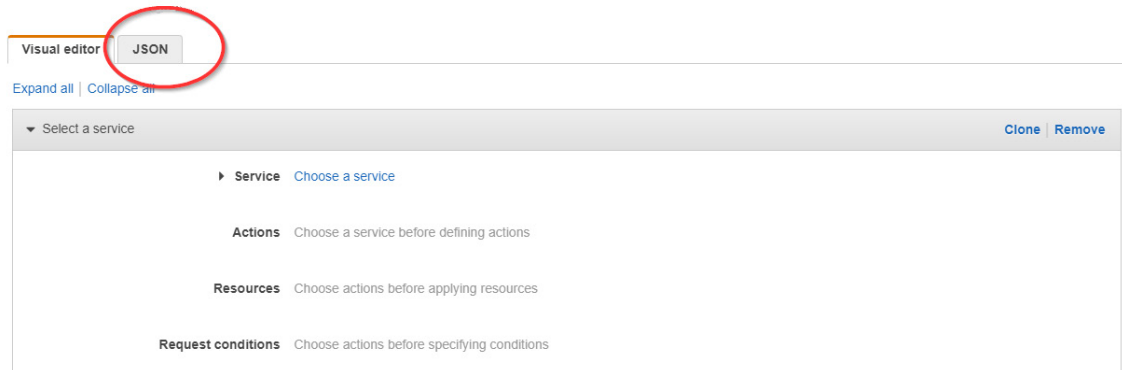
4. Follow the example in this step to see how to create a least privilege policy:
 - a. Copy the permissions for the Cost policy (see next page).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudCheckrCostPermissions",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeReservedInstancesOfferings",
        "ec2:DescribeReservedInstances",
        "ec2:DescribeReservedInstancesListings",
        "ec2:DescribeHostReservationOfferings",
        "ec2:DescribeReservedInstancesModifications",
        "ec2:DescribeHostReservations",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeRegions",
        "ec2:DescribeKeyPairs",
        "ec2:DescribePlacementGroups",
        "ec2:DescribeAddresses",
        "ec2:DescribeSpotInstanceRequests",
        "ec2:DescribeImages",
        "ec2:DescribeImageAttribute",
        "ec2:DescribeSnapshots",
        "ec2:DescribeVolumes",
        "ec2:DescribeTags",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeInstanceAttribute",
        "ec2:DescribeVolumeStatus",
        "elasticache:DescribeReservedCacheNodes",
        "elasticache:DescribeReservedCacheNodesOfferings",
        "rds:DescribeReservedDBInstances",
        "rds:DescribeReservedDBInstancesOfferings",
        "rds:DescribeDBInstances",
        "redshift:DescribeReservedNodes",
        "redshift:DescribeReservedNodeOfferings",
        "s3:GetBucketACL",
        "s3:GetBucketLocation",
        "s3:GetBucketLogging",
        "s3:GetBucketPolicy",
        "s3:GetBucketTagging",
        "s3:GetBucketWebsite",
        "s3:GetBucketNotification",
        "s3:GetLifecycleConfiguration",
        "s3:GetNotificationConfiguration",
        "s3:List*",
        "dynamodb:DescribeReservedCapacity",
        "dynamodb:DescribeReservedCapacityOfferings",
        "iam:GetAccountAuthorizationDetails",
        "iam:ListRolePolicies",
        "iam:ListAttachedRolePolicies"
      ],
      "Resource": "*"
    }
  ]
}

```

- b. Return to the Create Policy page in the AWS Management Console.
- c. Click the **JSON** tab.



The JSON tab opens, allowing you to create a policy using JSON syntax.

- d. Replace the text in the JSON tab with the policy you just copied.
 - e. Click **Review policy**. The Review policy page opens.
 - f. Type a name for the policy and click **Create policy**.
5. Repeat step 4 for each least privilege policy. You can copy these policies from [Appendix: IAM Policies](#).
 6. Copy the least privilege policy names to the [Required Information](#) section.

Attach Least Privilege Policies to Cross-Account Role

Now that you have created the least privilege policies, you must attach them to your cross-account role so that your self-hosted application can ingest the data from your AWS account.

1. From the AWS Services page, scroll down to the Security, Identity & Compliance section and select **IAM**.
The Welcome to Identity and Access Management screen displays.
2. From the dashboard, select **Policies**.
Select the radio button next to one of the least privilege policies. In this example, we chose **billing**.

| Filter policies <input type="text"/> | | Policy name <input type="text"/> | Type | Used as | Description |
|--------------------------------------|---|----------------------------------|------------------|------------------------|-------------|
| <input checked="" type="radio"/> | ▶ | billing | Customer managed | Permissions policy (3) | |
| <input type="radio"/> | ▶ | cloudtrail | Customer managed | Permissions policy (3) | |
| <input type="radio"/> | ▶ | cloudwatch | Customer managed | Permissions policy (3) | |
| <input type="radio"/> | ▶ | cost | Customer managed | Permissions policy (3) | |
| <input type="radio"/> | ▶ | inventory | Customer managed | Permissions policy (3) | |
| <input type="radio"/> | ▶ | security | Customer managed | Permissions policy (3) | |

3. From the Policy actions menu, select **Attach**.
The Attach Policy page opens.
4. Filter by your cross-account role name to refine your search, select the radio button next to your role, and click **Attach policy**.
5. Repeat steps 2-4 for the Cost, Inventory, Security/Compliance, CloudWatch Flow Logs, and CloudTrail policies.

Upgrade the Self-Hosted App

In this procedure, you will learn how to upgrade your self-hosted application.

Your first step is to create a snapshot or backup of your existing EBS volume in AWS and create a volume based on that snapshot. Since your EBS volume is essentially your D: drive, you will want to keep that data in case you need to restore or keep your original EC2 running.

1. From the EC2 list in AWS, select your EC2 instance.
2. Scroll down to the tabbed section. The Description tab of the selected instance is displayed by default.
3. Locate the link to the **xvdf** volume in the Block devices field.

| Description | | Status Checks | Monitoring | Tags |
|------------------------|--|---------------|------------|------|
| Instance ID | i-0211bc02 | | | |
| Instance state | stopped | | | |
| Instance type | m4.xlarge | | | |
| Elastic IPs | | | | |
| Availability zone | us-east-1b | | | |
| Security groups | launch-wizard-8. view inbound rules | | | |
| Scheduled events | - | | | |
| AMI ID | Cannot load details for ami-1da4f01a. You may not be permitted to view it. | | | |
| Platform | windows | | | |
| IAM role | CloudCheckr-InstanceProfile-Role | | | |
| Key pair name | S3SecretS3EAGIT | | | |
| Owner | 84989483003 | | | |
| Launch time | November 28, 2016 at 8:58:08 AM UTC-5 (7:54 hours) | | | |
| Termination protection | True | | | |
| Lifecycle | normal | | | |
| Public DNS (IPv4) | - | | | |
| IPv4 Public IP | - | | | |
| IPv6 IPs | - | | | |
| Private DNS | ip-20-04-80.ec2.internal | | | |
| Private IPs | 20.0.0.0 | | | |
| Secondary private IPs | | | | |
| VPC ID | vpc-f832e81d | | | |
| Subnet ID | subnet-6d8d172d | | | |
| Network interfaces | eth0 | | | |
| Source/dest. check | True | | | |
| ClassicLink | - | | | |
| EBS-optimized | True | | | |
| Root device type | ebs | | | |
| Root device | /dev/sda1 | | | |
| Block devices | /dev/sda1 xvdf | | | |

4. Click **xvdf**.
Information for the block device displays.
5. In the EBS ID field, click the link to the volume.

Block Device xvdf

| | |
|-----------------------|---|
| EBS ID | vol- 81bc021a-3c50-4000-8000-000000000000 |
| Root device type | EBS |
| Attachment time | 2017-11-28 12:17:44.0000000Z |
| Block device status | attached |
| Delete on termination | False |

The details of the selected volume are now displayed on the page.

6. Right-click the row of the selected volume and from the fly-out menu, select **Create Snapshot**.

The Create Snapshot page opens.

Volumes > Create Snapshot

Create Snapshot

Volume ⓘ

Description ⓘ

Encrypted Not Encrypted ⓘ

Key (127 characters maximum) Value (255 characters maximum)

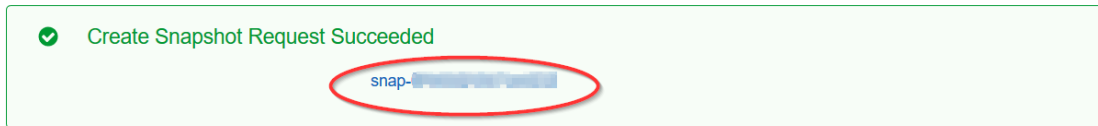
This resource currently has no tags

Choose the Add tag button or [click to add a Name tag](#)

Add Tag 50 remaining (Up to 50 tags maximum)

* Required Cancel **Create Snapshot**

7. Click **Create Snapshot**.
8. Click the **snapshot link**.



9. Right-click the row of the selected snapshot and from the fly-out menu, select **Create Volume**.

The Create Volume page opens.

Snapshots > Create Volume

Create Volume

Snapshot ID

Volume Type ⓘ

Size (GIB) (Min: 1 GiB, Max: 16384 GiB) ⓘ

IOPS 1500 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS) ⓘ

Availability Zone* ⓘ

Throughput (MB/s) Not applicable ⓘ

Encryption Not Encrypted

Key (127 characters maximum) Value (255 characters maximum)

This resource currently has no tags

Choose the Add tag button or [click to add a Name tag](#)

Add Tag 50 remaining (Up to 50 tags maximum)

* Required Cancel **Create Volume**

10. Click **Create volume**.
11. From the EC2 dashboard, select **Images > AMI**.
12. Select the check box next to the new AMI and click **Launch**.
13. Complete Step 2: Choose an Instance Type and Step 3: Configure Instance Details using the previous instructions.
14. In Step 3: Storage, click **X** to delete the original volume and click **Add New Volume**.

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 | Device | Snapshot | Size (GiB) | Volume Type | IOPS | Throughput (MB/s) | Delete on Termination | Encrypted |
|-------------|-----------|----------|------------|---------------------------|-------------|-------------------|-------------------------------------|---------------|
| Root | /dev/sda1 | | 30 | General Purpose SSD (gp2) | 100 / 3000 | N/A | <input checked="" type="checkbox"/> | Not Encrypted |
| EBS | xvdf | | 500 | General Purpose SSD (gp2) | 1500 / 3000 | N/A | <input type="checkbox"/> | Not Encrypted |

Add New Volume ← X →

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.

[Cancel](#)
[Previous](#)
[Review and Launch](#)
[Next: Add Tags](#)

A new row opens where you can add the copy of your original volume.

15. From the Device drop-down menu, select **xvdf** and copy the snapshot ID to replicate your original volume.
16. Complete the configuration of your new EC2 instance.
17. Remote into your new EC2 instance and complete the installation process.
18. Request a new license from your sales representative and follow the [License the App](#) section to upload the new license file.

Required Information

| Attribute | Value |
|---|-------------------------------------|
| AWS Account #1 | name/availability zone |
| AWS Account #2 | name/availability zone |
| AWS Account #3 | name/availability zone |
| Pricing Policy Name | |
| Pricing User Group Name | |
| Pricing User #1 (credentials for pricing jobs) | IAM user name/access key/secret key |
| Pricing User #2 (credentials for pricing jobs) | IAM user name/access key/secret key |
| Pricing User #3 (credentials for pricing jobs) | IAM user name/access key/secret key |
| EC2 Instance ID | |
| EC2 Instance Type | |
| EC2 Availability Zone (Region Code) | |
| Private Key (.PEM) File Location and Name | |
| Public DNS Name (IPv4) | |
| Private DNS | |
| Subnet ID | |
| Trusted User Policy Name | |
| Trusted User Group Name | |

| Attribute | Value |
|--|-------|
| Trusted User Name | |
| Cross-Account Role Name | |
| Least Privileges Policy: Cost | |
| Least Privileges Policy: Billing | |
| Least Privileges Policy: Security/Compliance | |
| Least Privileges Policy: Inventory | |
| Least Privileges Policy: CloudWatch Logs | |
| Least Privileges Policy: CloudTrail | |
| Partner Name | |
| Account Name(s) | |

Frequently Asked Questions

Is There an Alternative to Remote Desktop?

If you want to connect to your EC2 instance on your local machine, you can use the external public hostname to connect to the EC2 instance to install the application.

The external public hostname resolves to the public IP address or the Elastic IP address, which allows your instance to communicate to the internet.

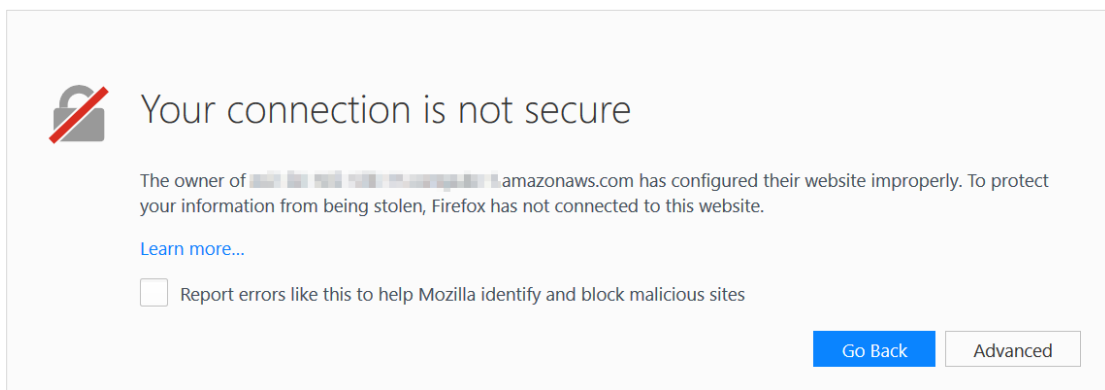
| Description | |
|-------------------|--|
| Instance ID | i-00d309a083e4f0435 |
| Instance state | |
| Instance type | t3.small |
| Elastic IPs | 0.00.000.000* |
| Public DNS (IPv4) | ec2-0-00-000-000.compute-1.amazonaws.com |
| IPv4 Public IP | 0.00.000.000 |

1. Open a Web browser. This procedure uses Mozilla Firefox as an example.
2. Click + to open a new tab.
3. In the address bar, type **http://**
4. Paste the public **DNS (IPv4)** into the address bar.
5. Add **:8080/** to the end of the host name to allow the Web installer to run on port 8080 in HTTP. You opened this port as part of your security group configuration.

The format of the complete address will look like this:



6. Click **Enter**.
The first screen of the Web installer opens.
7. Complete the installation steps for the Web Console in the [Install the Self-Hosted App](#) section.
When the configuration is complete, a warning message indicates that your connection is not secure.

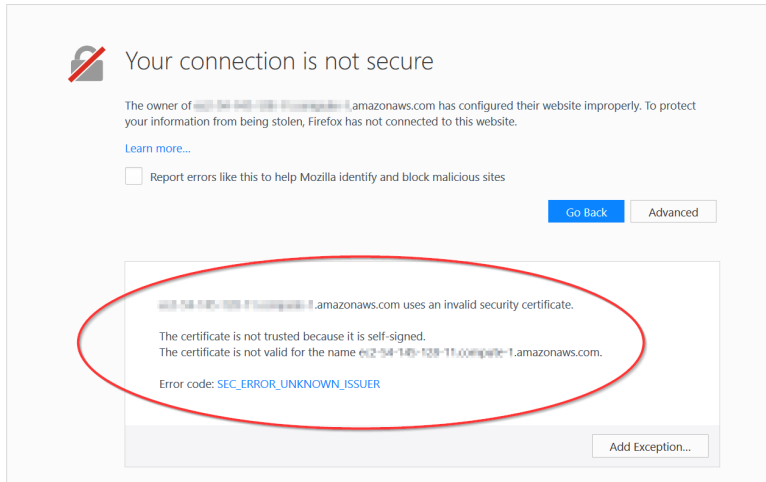


Note: The content and look-and-feel of the warning message depends on the browser in use. In this example, we used Mozilla Firefox.

The application requires a secure connection with a certificate owned by the domain. Since you are launching the application in a self-hosted environment, it cannot automatically create a certificate.

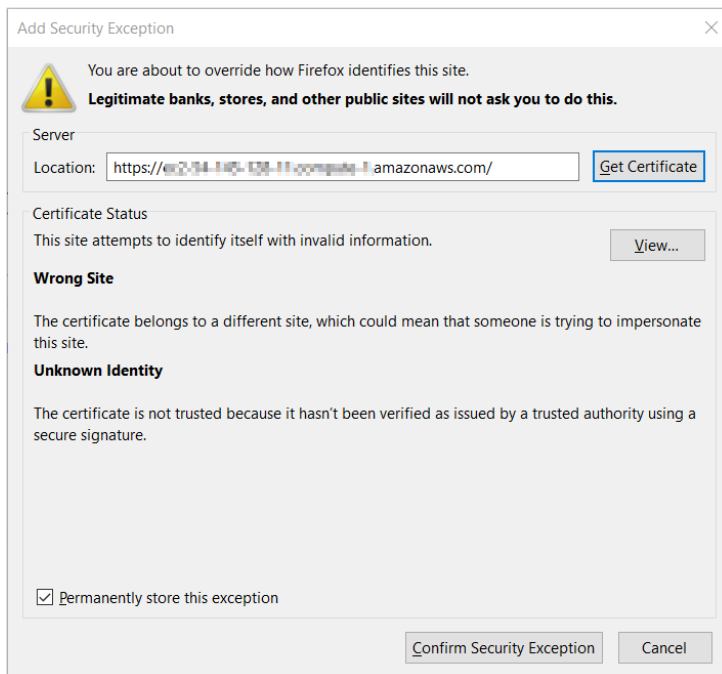
8. Click **Advanced** to get more information about the warning.

A message indicates that the certificate is not trusted or valid.



9. Click **Add Exception...** to add the EC2 instance as a security exception.

The Add Security Exception dialog box opens.

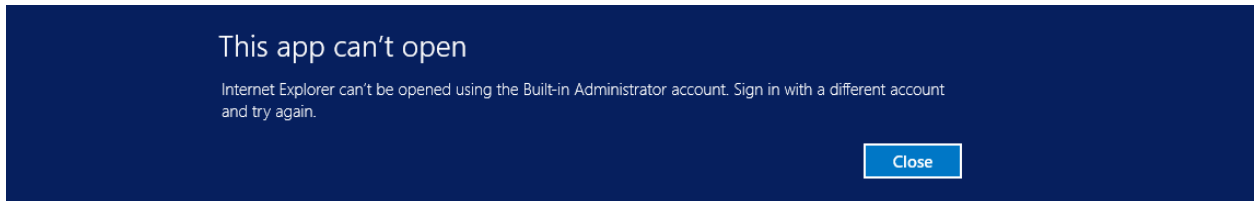


10. Verify that **Permanently store this exception** is selected and click **Confirm Security Exception**.

The log in screen of the application opens.

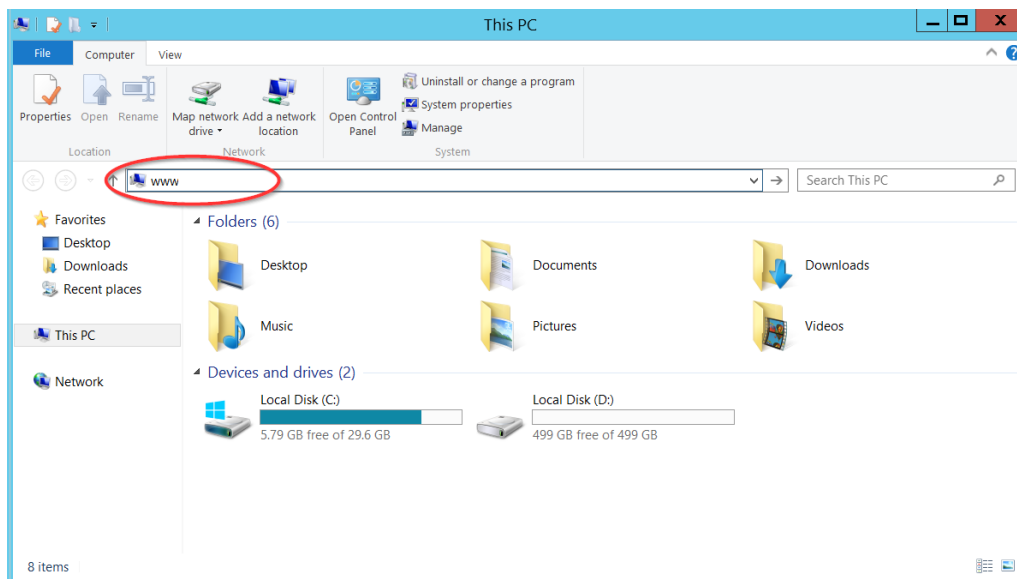
Why Can't I Open My Browser?

Your remote desktop session runs in a Microsoft Windows® 2012 R2 server environment, which is not compatible with newer applications like Internet Explorer. As a result, since you are logging in as an administrator, you will get an error message when you select **Start > Internet Explorer**:



Here is the workaround to open a browser session:

1. From the taskbar, click the **Folder** icon.
2. Type **www** in the search bar to open your browser.

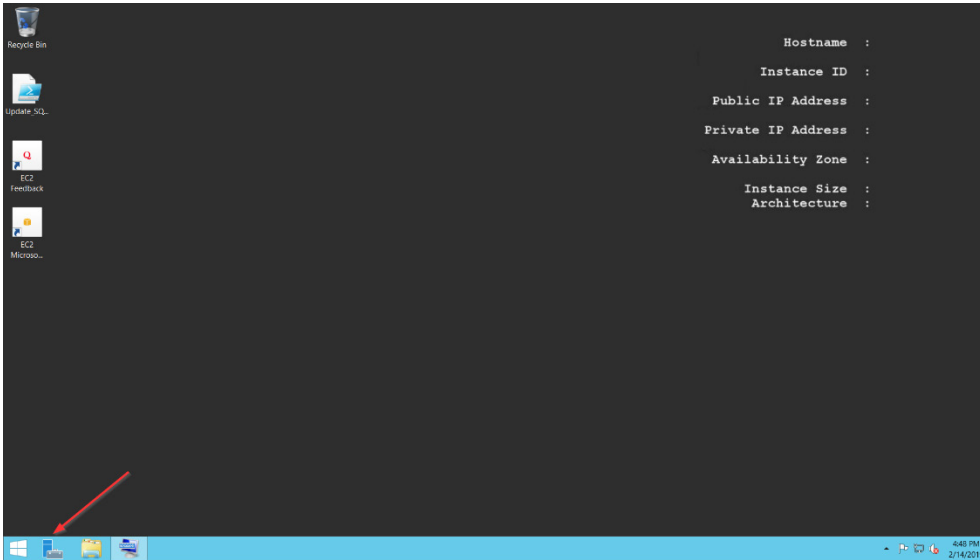


3. Follow steps 11-17 in the [Install the Self-Hosted App](#) section to complete your connection to your EC2 instance.

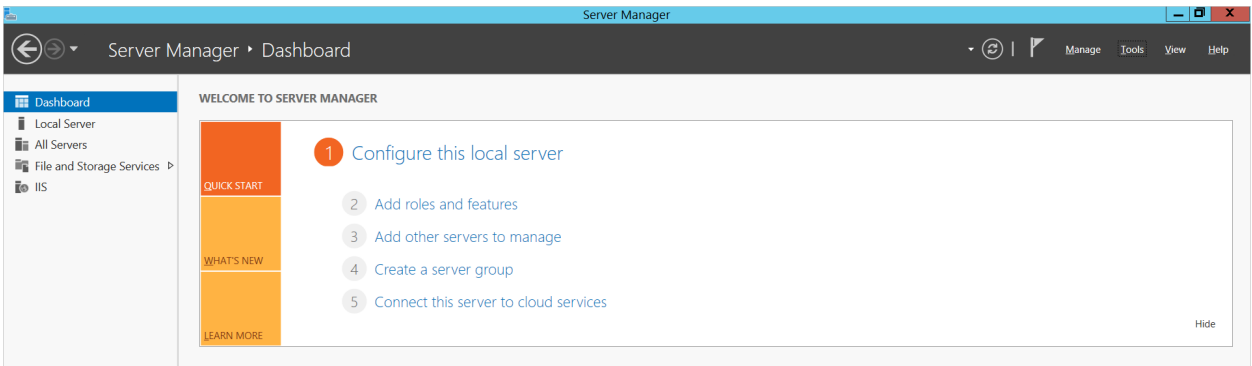
Where Is My D: Drive?

If your D: drive seems to be missing, follow these steps to make sure it is online:

1. From the taskbar, click the **Server Manager** icon.

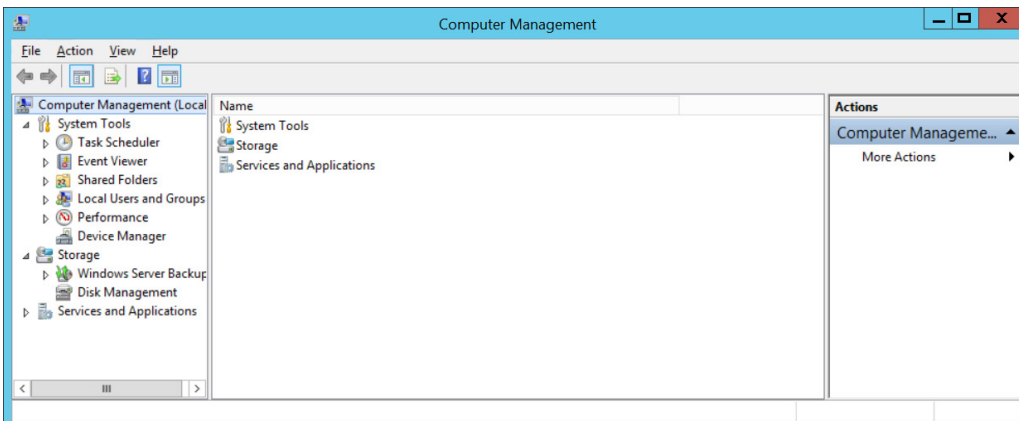


The Server Manager Dashboard opens.

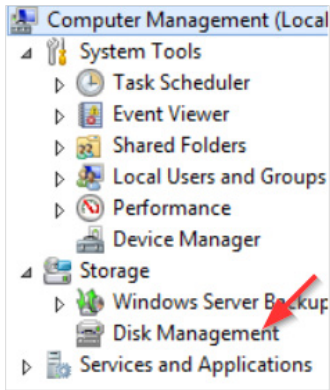


2. From the menu bar, choose **Tools > Computer Management**.

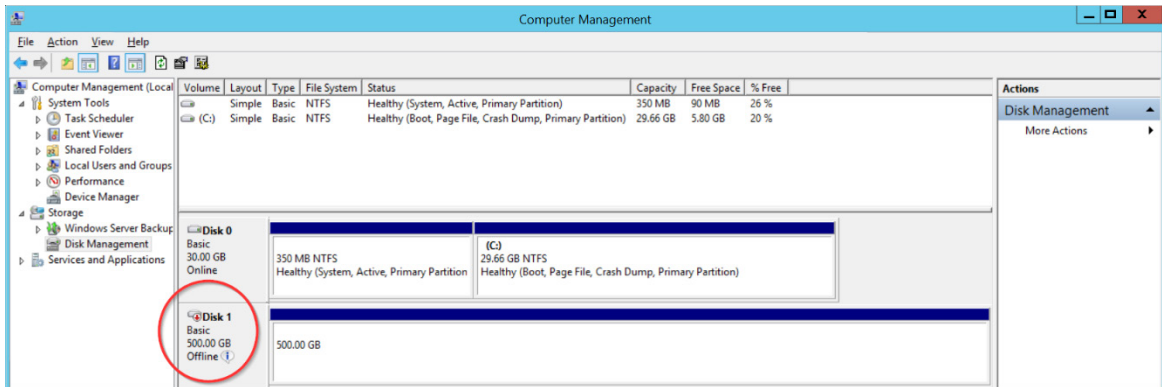
The Computer Management screen displays.



- From the dashboard, select **Storage > Disk Management**.



Information about your disks displays. Notice that Disk 1 has a red arrow and is labeled **Offline**.

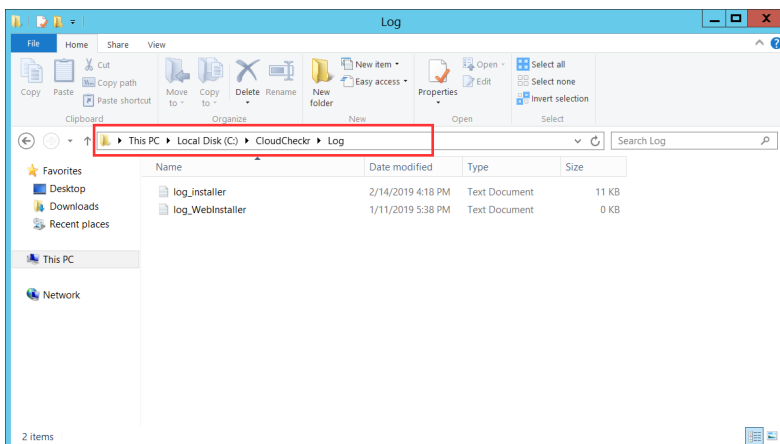


- Right-click the **disk name** and from the fly-out menu, select **Online**. Your D: drive is now available.

How Do I Access My Log Files?

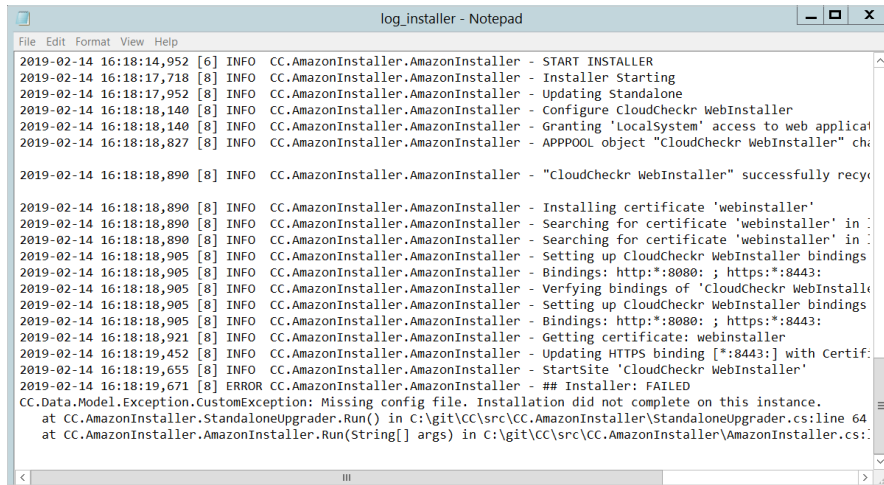
Our team can help you diagnose and solve the problem by reviewing your **log files**, which record every action performed within the web installer and the application. Follow these steps to access your log files:

- From the taskbar, click **Windows Explorer**.
- Navigate to **PC: Local C > CloudCheckr > Logs**.



3. Click one of the log files.

In this example, we opened the log file for the application installer.



```
log_installer - Notepad
File Edit Format View Help
2019-02-14 16:18:14,952 [6] INFO CC.AmazonInstaller.AmazonInstaller - START INSTALLER
2019-02-14 16:18:17,718 [8] INFO CC.AmazonInstaller.AmazonInstaller - Installer Starting
2019-02-14 16:18:17,952 [8] INFO CC.AmazonInstaller.AmazonInstaller - Updating Standalone
2019-02-14 16:18:18,140 [8] INFO CC.AmazonInstaller.AmazonInstaller - configure CloudCheckr WebInstaller
2019-02-14 16:18:18,140 [8] INFO CC.AmazonInstaller.AmazonInstaller - Granting 'localSystem' access to web applicat
2019-02-14 16:18:18,827 [8] INFO CC.AmazonInstaller.AmazonInstaller - APPPOOL object "CloudCheckr WebInstaller" ch
2019-02-14 16:18:18,890 [8] INFO CC.AmazonInstaller.AmazonInstaller - "CloudCheckr WebInstaller" successfully recy
2019-02-14 16:18:18,890 [8] INFO CC.AmazonInstaller.AmazonInstaller - Installing certificate 'webinstaller'
2019-02-14 16:18:18,890 [8] INFO CC.AmazonInstaller.AmazonInstaller - Searching for certificate 'webinstaller' in :
2019-02-14 16:18:18,890 [8] INFO CC.AmazonInstaller.AmazonInstaller - Searching for certificate 'webinstaller' in :
2019-02-14 16:18:18,905 [8] INFO CC.AmazonInstaller.AmazonInstaller - Setting up CloudCheckr WebInstaller bindings
2019-02-14 16:18:18,905 [8] INFO CC.AmazonInstaller.AmazonInstaller - Bindings: http*:8080: ; https*:8443:
2019-02-14 16:18:18,905 [8] INFO CC.AmazonInstaller.AmazonInstaller - Verifying bindings of 'CloudCheckr WebInstall
2019-02-14 16:18:18,905 [8] INFO CC.AmazonInstaller.AmazonInstaller - Setting up CloudCheckr WebInstaller bindings
2019-02-14 16:18:18,905 [8] INFO CC.AmazonInstaller.AmazonInstaller - Bindings: http*:8080: ; https*:8443:
2019-02-14 16:18:18,921 [8] INFO CC.AmazonInstaller.AmazonInstaller - Getting certificate: webinstaller
2019-02-14 16:18:19,452 [8] INFO CC.AmazonInstaller.AmazonInstaller - Updating HTTPS binding [*:8443:] with Certif:
2019-02-14 16:18:19,655 [8] INFO CC.AmazonInstaller.AmazonInstaller - Startsite 'CloudCheckr WebInstaller'
2019-02-14 16:18:19,671 [8] ERROR CC.AmazonInstaller.AmazonInstaller - ## Installer: FAILED
CC.Data.Model.Exception.CustomException: Missing config file. Installation did not complete on this instance.
   at CC.AmazonInstaller.StandaloneUpgrader.Run() in C:\git\CC\src\CC.AmazonInstaller\StandaloneUpgrader.cs:line 64
   at CC.AmazonInstaller.AmazonInstaller.Run(String[] args) in C:\git\CC\src\CC.AmazonInstaller\AmazonInstaller.cs:
```

4. Scroll down the bottom of the list to see the most recent events.
5. Provide the log file or a screenshot of that log file to Support so they can troubleshoot your issue.

Appendix

IAM Policies

If you are going to [create a cross-account role](#) manually, you can copy any of these least privilege policy categories and attach them to that role.

Billing:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CostReadDBR",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketACL",
        "s3:GetBucketLocation",
        "s3:GetBucketLogging",
        "s3:GetBucketPolicy",
        "s3:GetBucketTagging",
        "s3:GetBucketWebsite",
        "s3:GetBucketNotification",
        "s3:GetLifecycleConfiguration",
        "s3:GetNotificationConfiguration",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::[YOUR DETAILED BILLING REPORT BUCKET]",
        "arn:aws:s3:::[YOUR DETAILED BILLING REPORT BUCKET]/*",
        "arn:aws:s3:::[YOUR COST AND USAGE REPORT BUCKET] (optional)",
        "arn:aws:s3:::[YOUR COST AND USAGE REPORT BUCKET]/* (optional)"
      ]
    }
  ]
}
```

Security/Compliance:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SecurityPermissions",
      "Effect": "Allow",
      "Action": [
        "acm:DescribeCertificate",
        "acm:ListCertificates",
        "acm:GetCertificate",
        "cloudtrail:DescribeTrails",
        "cloudtrail:GetTrailStatus",
        "logs:GetLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "config:DescribeConfigRules",
        "config:GetComplianceDetailsByConfigRule",
        "config:DescribeDeliveryChannels",
        "config:DescribeDeliveryChannelStatus",
        "config:DescribeConfigurationRecorders",
        "config:DescribeConfigurationRecorderStatus",
        "ec2:Describe*",
        "iam:Get*",
        "iam:List*",
        "iam:GenerateCredentialReport",
        "kms:DescribeKey",
        "kms:GetKeyPolicy",
        "kms:GetKeyRotationStatus",
        "kms:ListAliases",
        "kms:ListGrants",
        "kms:ListKeys",
        "kms:ListKeyPolicies",
        "kms:ListResourceTags",
        "rds:Describe*",
        "ses:ListIdentities",
        "ses:GetSendStatistics",
        "ses:GetIdentityDkimAttributes",
        "ses:GetIdentityVerificationAttributes",
        "ses:GetSendQuota",
        "sns:GetSnsTopic",
        "sns:GetTopicAttributes",
        "sns:GetSubscriptionAttributes",
        "sns:ListTopics",
        "sns:ListSubscriptionsByTopic",
        "sqs:ListQueues",
        "sqs:GetQueueAttributes"
      ],
      "Resource": "*"
    }
  ]
}
```

Inventory:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SecurityPermissions",
      "Effect": "Allow",
      "Action": [
        "acm:DescribeCertificate",
        "acm:ListCertificates",
        "acm:GetCertificate",
        "cloudtrail:DescribeTrails",
        "cloudtrail:GetTrailStatus",
        "logs:GetLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams",
        "config:DescribeConfigRules",
        "config:GetComplianceDetailsByConfigRule",
        "config:DescribeDeliveryChannels",
        "config:DescribeDeliveryChannelStatus",
        "config:DescribeConfigurationRecorders",
        "config:DescribeConfigurationRecorderStatus",
        "ec2:Describe*",
        "iam:Get*",
        "iam:List*",
        "iam:GenerateCredentialReport",
        "kms:DescribeKey",
        "kms:GetKeyPolicy",
        "kms:GetKeyRotationStatus",
        "kms:ListAliases",
        "kms:ListGrants",
        "kms:ListKeys",
        "kms:ListKeyPolicies",
        "kms:ListResourceTags",
        "rds:Describe*",
        "ses:ListIdentities",
        "ses:GetSendStatistics",
        "ses:GetIdentityDkimAttributes",
        "ses:GetIdentityVerificationAttributes",
        "ses:GetSendQuota",
        "sns:GetSnsTopic",
        "sns:GetTopicAttributes",
        "sns:GetSubscriptionAttributes",
        "sns:ListTopics",
        "sns:ListSubscriptionsByTopic",
        "sqs:ListQueues",
        "sqs:GetQueueAttributes"
      ],
      "Resource": "*"
    }
  ]
}
```

CloudWatch Flow Logs:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudWatchLogsSpecific",
      "Effect": "Allow",
      "Action": [
        "logs:GetLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "arn:aws:logs:*:*:*"
      ]
    }
  ]
}
```

CloudTrail:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudTrailPermissions",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketACL",
        "s3:GetBucketLocation",
        "s3:GetBucketLogging",
        "s3:GetBucketPolicy",
        "s3:GetBucketTagging",
        "s3:GetBucketWebsite",
        "s3:GetBucketNotification",
        "s3:GetLifecycleConfiguration",
        "s3:GetNotificationConfiguration",
        "s3:GetObject",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::[YOUR CLOUDTRAIL BUCKET]",
        "arn:aws:s3:::[YOUR CLOUDTRAIL BUCKET]/*"
      ]
    }
  ]
}
```

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