Self-Hosted Application: Multi-Tier – Commercial

GETTING STARTED GUIDE

Version 16.5

CloudCheckr

Table of Contents

Introduction 4
Configure Services and Resources in AWS 5
Create an IAM Policy 5
Create an IAM User 8
Create an RDS Database 11
Create an S3 Bucket 15
Create an S3 Bucket Policy 16
Create an IAM Role 18
Launch the Self-Hosted AMI 20
Configure the EC2 Instances 21
Configure the Web Console 21
Configure the Scheduler and Workers 26
Install the Self-Hosted App 27
Install the Web Console 27
Input JSON File 37
Install the Scheduler 39
Install the Workers 39
Configure the Self-Hosted App 40
License Your App 40
Create a Partner 42
Complete the Back-End System Configuration 45
Create a Trusted User 47
Create Trusted User Not in a Standard Region 48
Create an Account 49
Create Least Privilege Policies 51
Create a Cross-Account Role 51
Upgrade the Self-Hosted App 55
Frequently Asked Questions 57
Is There an Alternative to Remote Desktop? 57
Why Can't I Open My Browser? 59

Where Is My D: Drive? 60
How Do I Access My Log Files? 62
Required Information 64
Appendix 66
IAM Policies 66
Cost 67
Billing: DBR 68
Billing: CUR 68
Security/Compliance 69
Inventory (code block 1 of 3) 70
Inventory (code block 2 of 3) 71
Inventory (code block 3 of 3) 72
CloudTrail 73
CloudWatch Flow Logs 73
Deploy CloudCheckr in Additional Availability Zone 74

INTRODUCTION

This guide describes how to configure the **Multi-Tier – Commercial (MT-COMM) self-hosted application**, which launches CloudCheckr in a virtual private cloud (VPC) where your data and security is completely protected.

The MT-COMM version is well-suited for customers who need a scalable architecture for larger cloud deployments. Although the configuration is a bit more complex, it allows customers to spread their workload across multiple servers for greater stability.

You will need to configure three separate Elastic Compute Cloud (EC2) instances:

- an EC2 instance for the **Web Console**: where you will log in and use the application
- two EC2 instances, one for the **Scheduler** and one for the Workers, which are the background processes that collect and store your AWS data

In addition, you will need to configure:

- a **Relational Database Service (RDS)**, a Microsoft SQL® server database that stores your data
- an **S3 bucket**, a private S3 bucket that houses encryption keys and other storage data
- **IAM role(s)**: AWS identities that allow you to access your S3 bucket and AWS account(s)

We recommend that you record key information generated during your AWS configuration to the Required Information section. You will need this information for your CloudCheckr setup and for troubleshooting. Items you may wish to record 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.

Your first step is to create **three** AWS Identity and Access Management (IAM) users. AWS will generate a unique access key and secret key for each user. When you plug these keys into CloudCheckr, 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 enables the self-hosted application to access AWS pricing data
- create an IAM user and attach them to your pricing policy

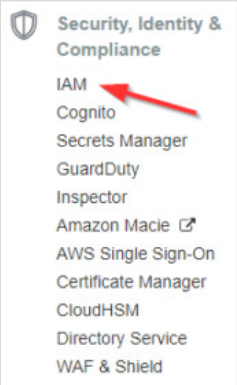
To ensure that your self-hosted application contains a good cross-section of availability zones and pricing data, you must create each IAM user in **three separate AWS accounts**.

Create an IAM Policy

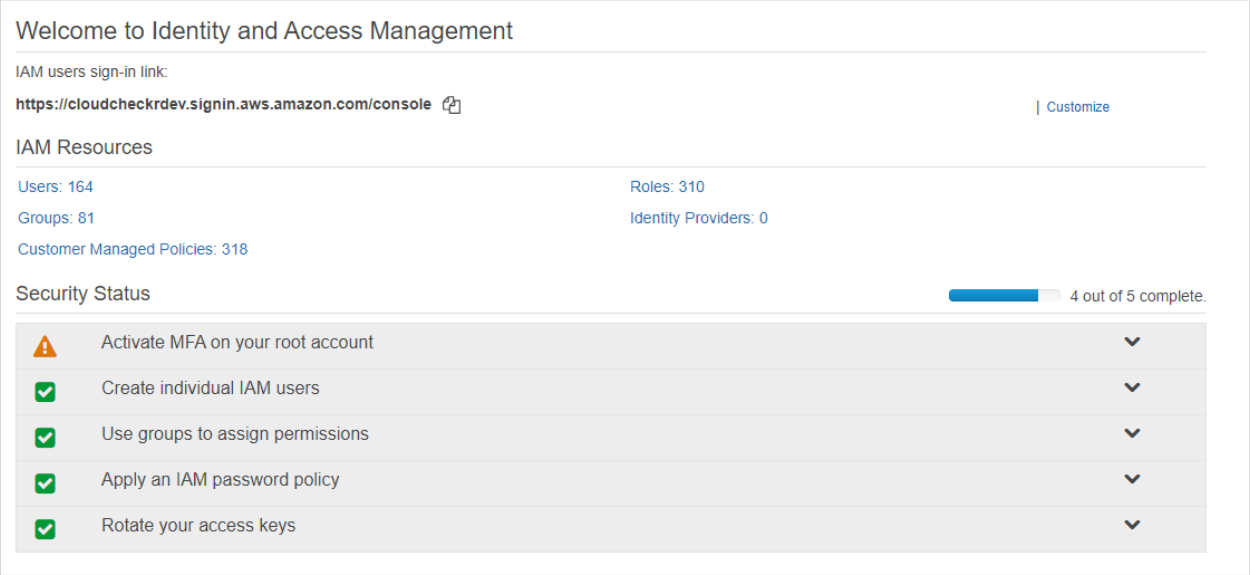
In this procedure, you will create an IAM policy that will give the self-hosted application the permissions it needs to access the AWS pricing data.

1. Launch the AWS Management Console associated with your first AWS account.

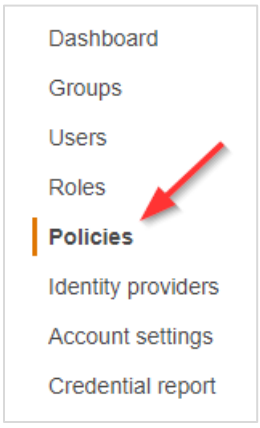
2. On the AWS Services page, scroll down to Security, Identity & Compliance and select **IAM**.



The Welcome to Identity and Access Management screen displays.



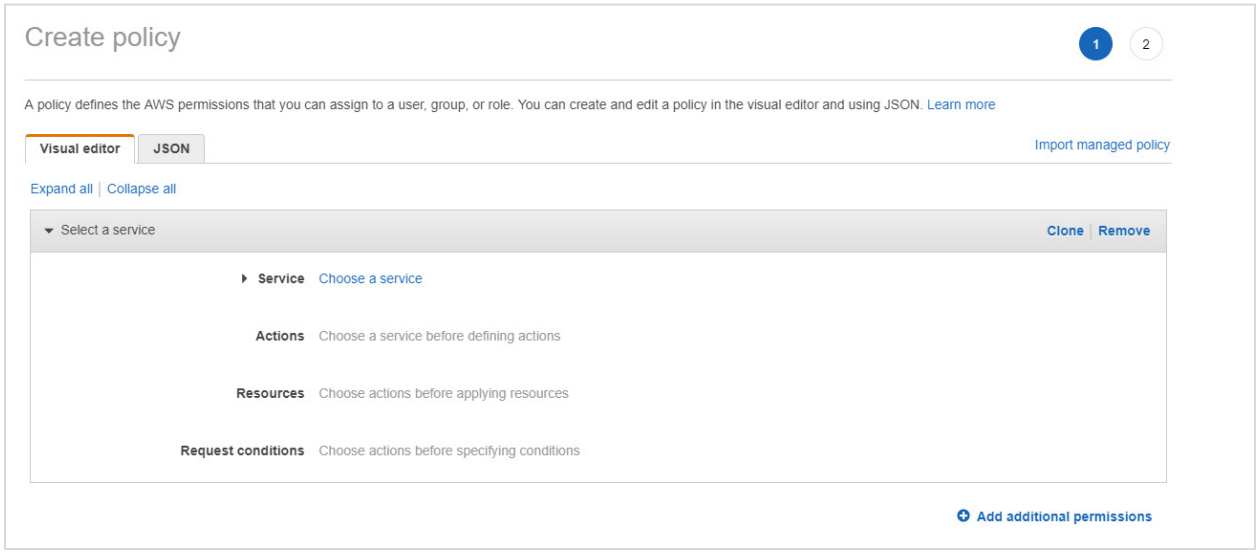
3. From the dashboard, click **Policies**.



A list of policies displays.

4. Click **Create policy**.

The Create Policy page opens.

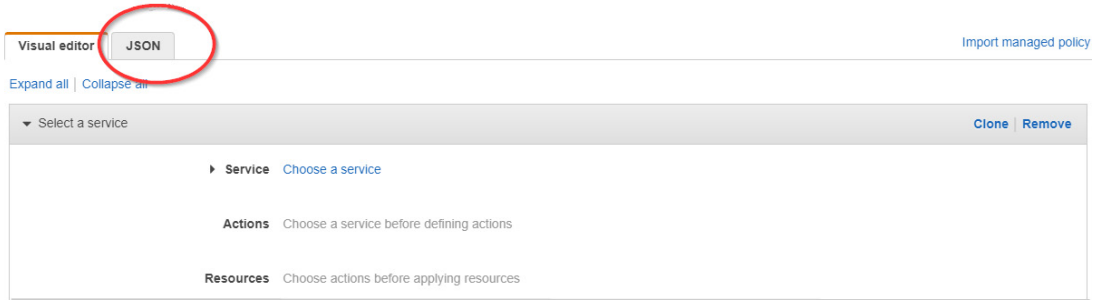


5. Follow the example in this step to see how to create the pricing policy:

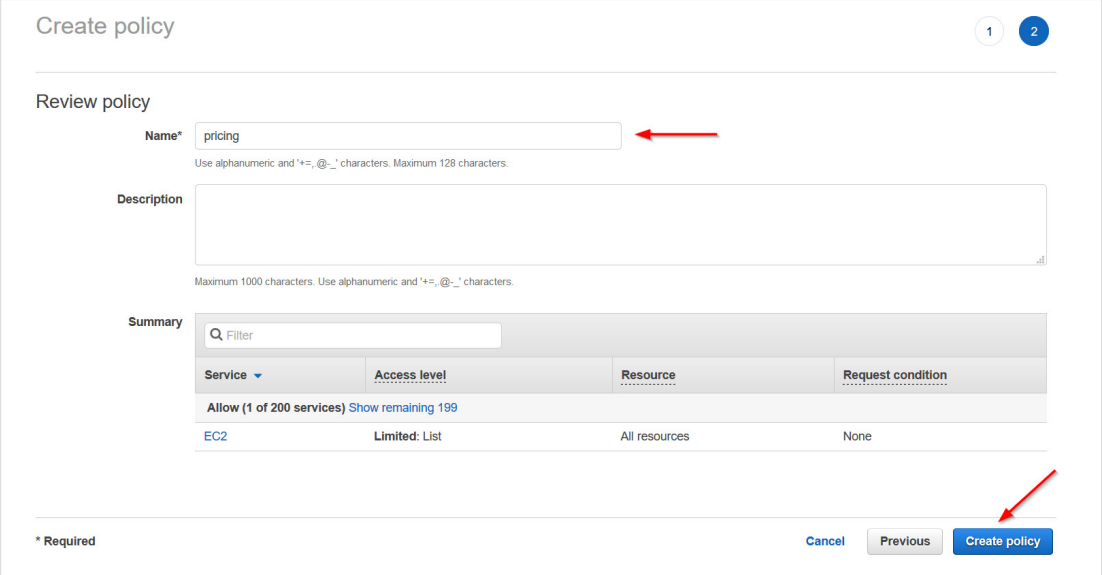
    a. Copy this pricing policy to your clipboard:

```
{
    "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.



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**.



A message indicates that AWS has created your policy.

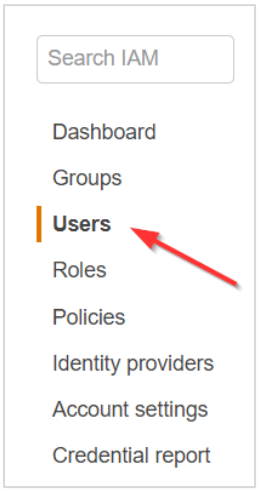
6. Repeat steps 1-5 to create a pricing policy for the remaining two AWS accounts.

7. Copy the names of each pricing policy to the Required Information section.

Create an IAM User

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

1. Return to the IAM dashboard and click **Users**.

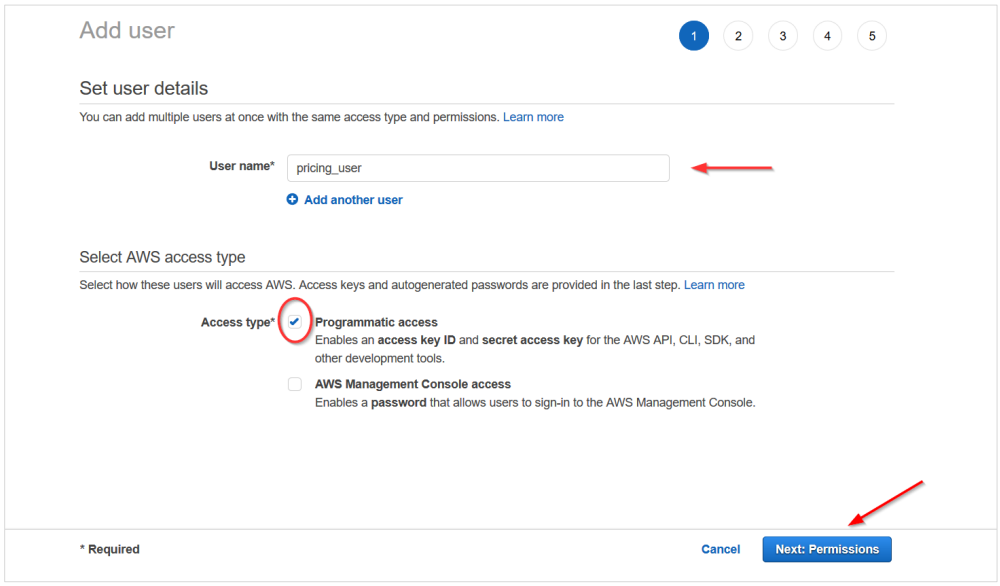


A list of users displays.

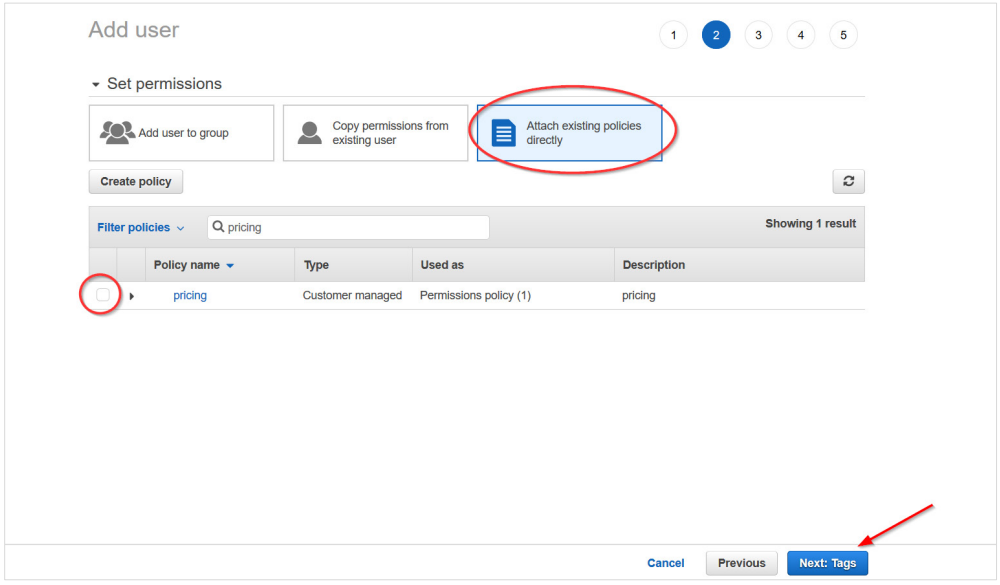
2. Click **Add user**. The Add User wizard opens.

3. On this screen:

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



4. Click **Attach existing policies directly**, select your pricing policy, and click **Next: Tags**.

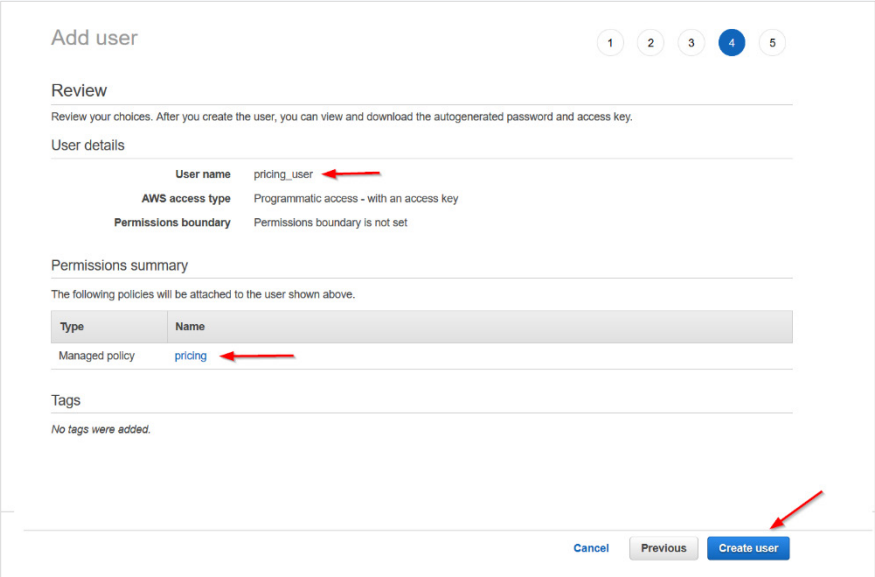


The optional Add tags page displays. For the purposes of this procedure, we will not add tags.

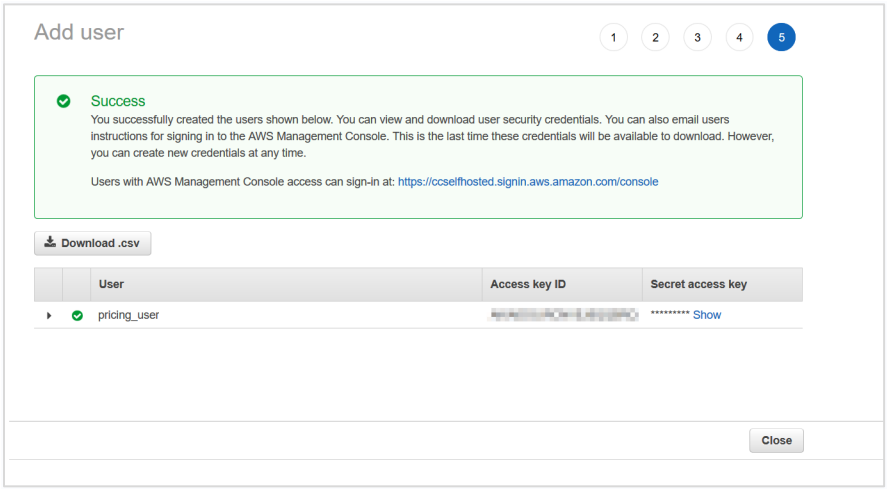
5. Click **Next: Review**.

   This page displays the name of your user and verifies that you attached the pricing policy to them.

6. Click **Create user**.



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



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

   Note: This is the only time you can download or copy these keys. If you misplace them, you can create new keys. See Resetting Your Lost or Forgotten Passwords or Access Keys for details.

8. Repeat steps 1-7 for the remaining two AWS accounts.

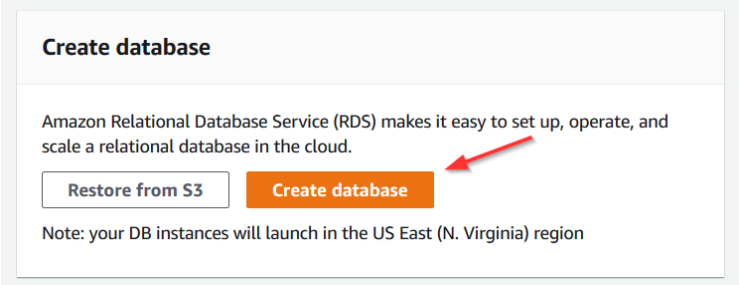
9. For each of the three IAM users, copy the username, access key, and secret key to the Required Information section.

Create an RDS Database

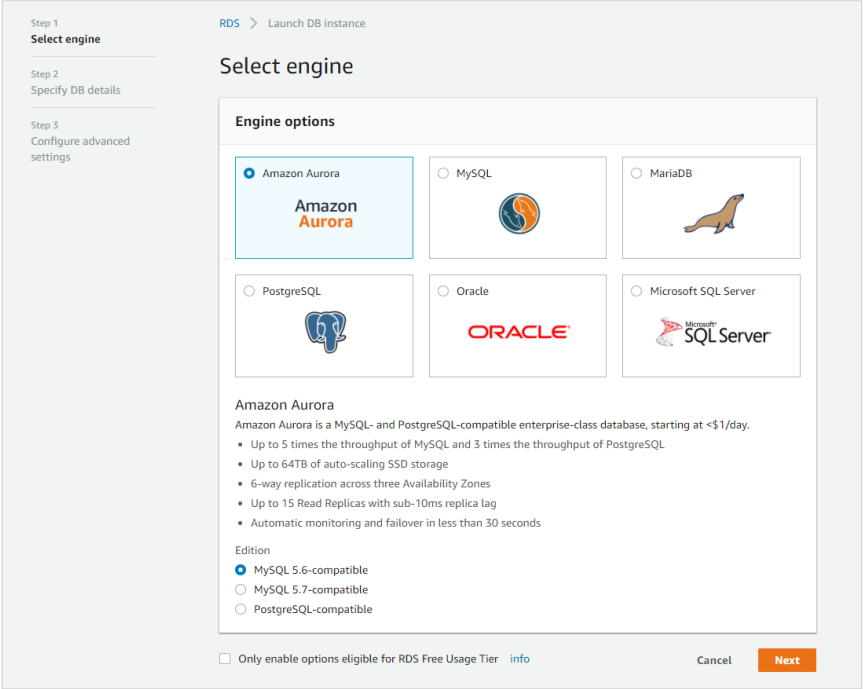
You need to create an RDS database that will store your data. You must set up the database before installation because the Web installer will prompt you to connect to the database during installation.

1. Return to the AWS Services page.
2. From the Database section, select **RDS**.

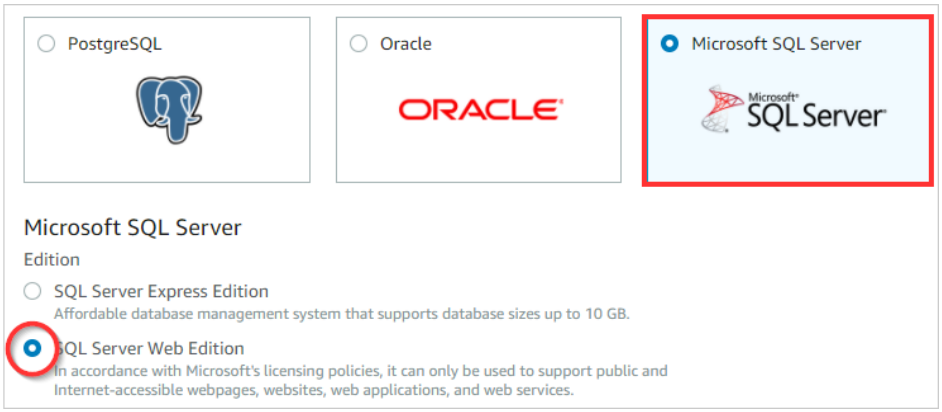
   The Amazon RDS page opens.
3. Navigate to the Create database section and click **Create database**.



The Select engine wizard opens.



4. Select **Microsoft SQL Server and SQL Server Web Edition**.

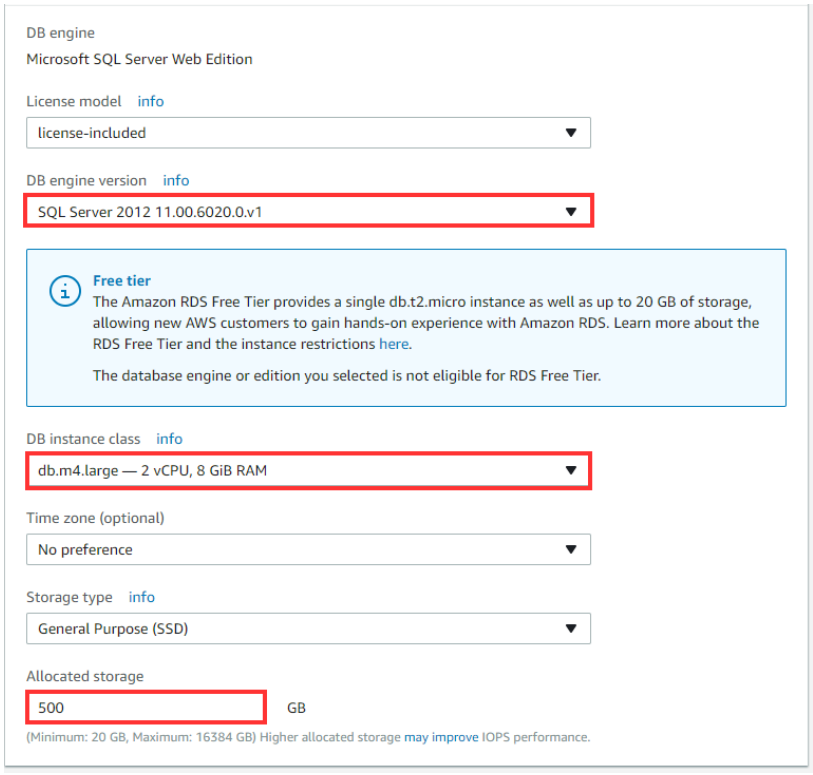


5. Click **Next**. The Specify DB details page opens.

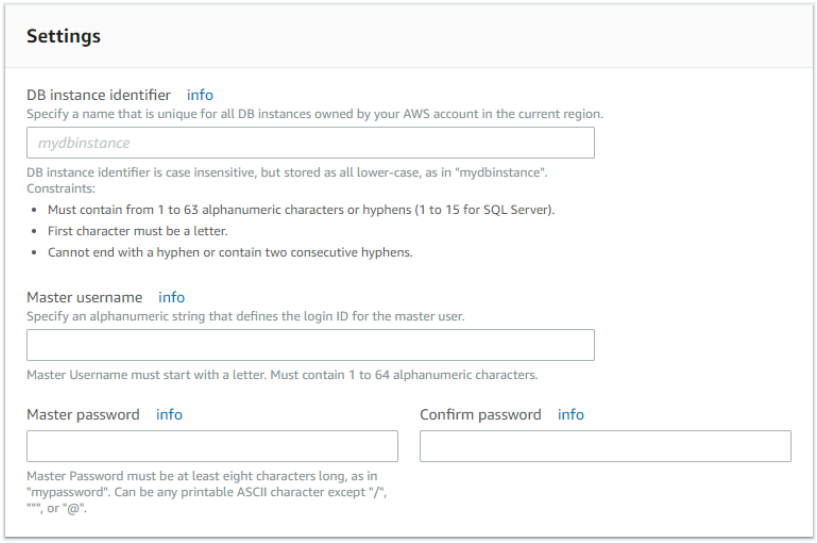
6. From the DB engine version drop-down menu, select **SQL Server 11.00.6020.0.v1**.

7. From the DB instance class drop-down menu, select **db.m4.large**.

8. Type **500** for the allocated storage.



9. Scroll down to the Settings section.

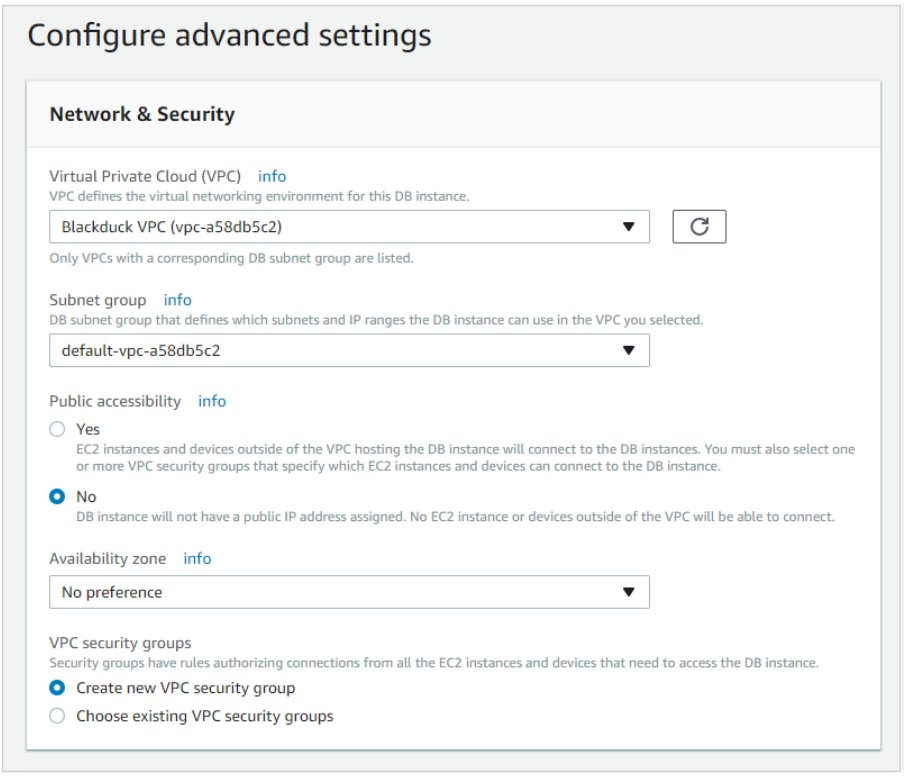


10. In the DB instance identifier text field, type a name for your RDS database.

11. Create a master username and password that you will use to connect to your RDS database.

12. Copy the database name, username, and password to the Required Information section.

13. Click **Next**. The Configure advanced settings page opens.



14. Configure your settings as follows:

    a. In the Virtual Private Cloud (VPC) section, select an appropriate VPC for your environment.

    b. In the Publicly Accessible section, select the **No** radio button.

    c. In the Availability zone section, leave the **No preference** selection.

    d. In the VPC security groups section, select the **Create new VPC security group** radio button. You will create the rules for this security group when you configure the EC2 instance.

    e. Talk to your IT department to find out if they want you to configure the remaining settings.

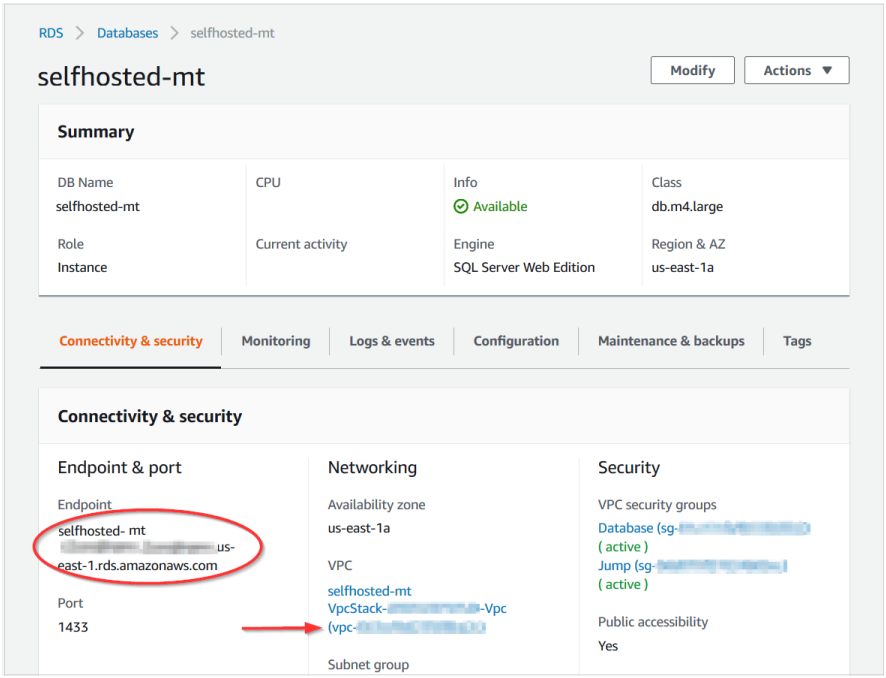
15. Click **Create database**.

16. Select your new database from the list.

    The details on the new database display.

    Under Endpoint & port, notice the endpoint value, which you will use as the RDS database.

    Under VPC, notice the VPC value, which is the VPC you will select during your EC2 configuration.



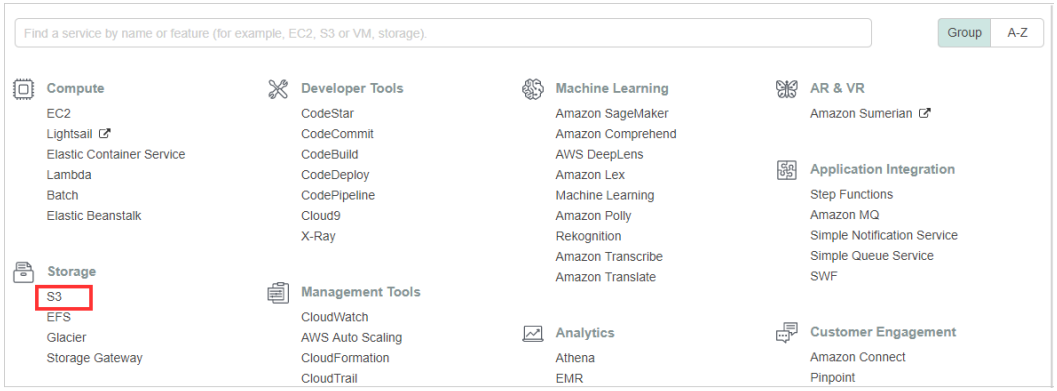
17. Copy the endpoint and VPC values to the Required Information section.

Create an S3 Bucket

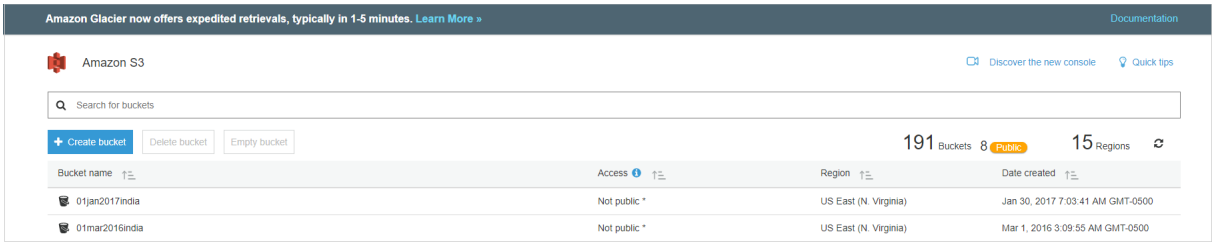
In this procedure, you will create an S3 bucket where AWS will store your encryption keys and data.

1. Log in to the AWS Management Console.

2. From the Storage section, select **S3**.



The Amazon S3 page opens.



3. Click **+ Create bucket**.

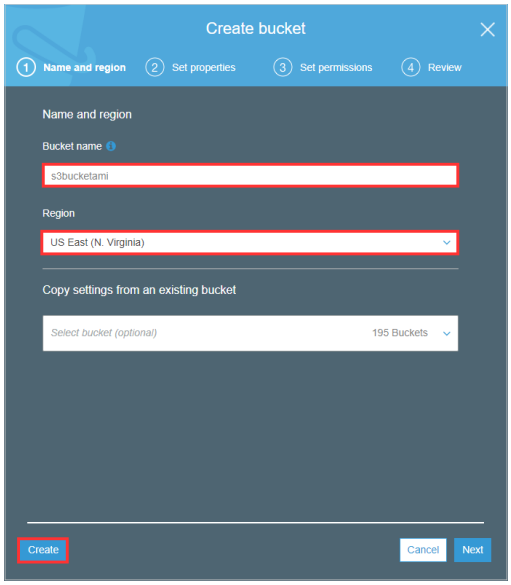
   The Create bucket configuration wizard opens.

4. Configure your S3 bucket:

    a. In the Bucket name text field, type a bucket name.

    b. From the Region drop-down menu, select a region.

    c. Click **Create**.



    The new S3 bucket is now displayed in the list.

5. Copy the S3 bucket name and region to the Required Information section.

Create an S3 Bucket Policy

In this procedure, you will create a policy that will give your EC2 instance permission to communicate with your S3 bucket.

1. Return to the AWS services page.

2. Scroll down to the Security, Identity & Compliance section and select **IAM**.

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

4. Click **Create policy**. The Create Policy page opens.

5. Follow the example in this step to see how to create the S3 bucket policy:

   a. Copy this S3 bucket policy to your clipboard:

```
{
        "Version":"2012-10-17",
        "Statement":[
           {
              "Sid":"Stmt1489160892000",
              "Effect":"Allow",
              "Action":[
                 "s3:*"
              ],
              "Resource":[
                 "arn:aws:s3:::bucket_name",
                 "arn:aws:s3:::bucket_name/key_name"
              ]
           }
        ]
     }
```

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

   c. Click the **JSON** tab.

   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**.

      A message indicates that AWS has created your policy.

      Note: The above policy is for MT – COMM only. For GovCloud, use the following policy:

```
{
        "Version":"2012-10-17",
        "Statement":[
           {
              "Sid":"Stmt1489160892000",
              "Effect":"Allow",
              "Action":[
                 "s3:*"
              ],
              "Resource":[
                 "arn:aws-us-gov:s3:::bucket_name",
                 "arn:aws-us-gov:s3:::bucket_name/key_name"
              ]
           }
        ]
     }
```

Note: Both the MT – COMM and MT – COMM (GovCloud) use a general ARN format since ARNs may be formatted differently across regions.

Use the following guidelines to modify the ARNs:

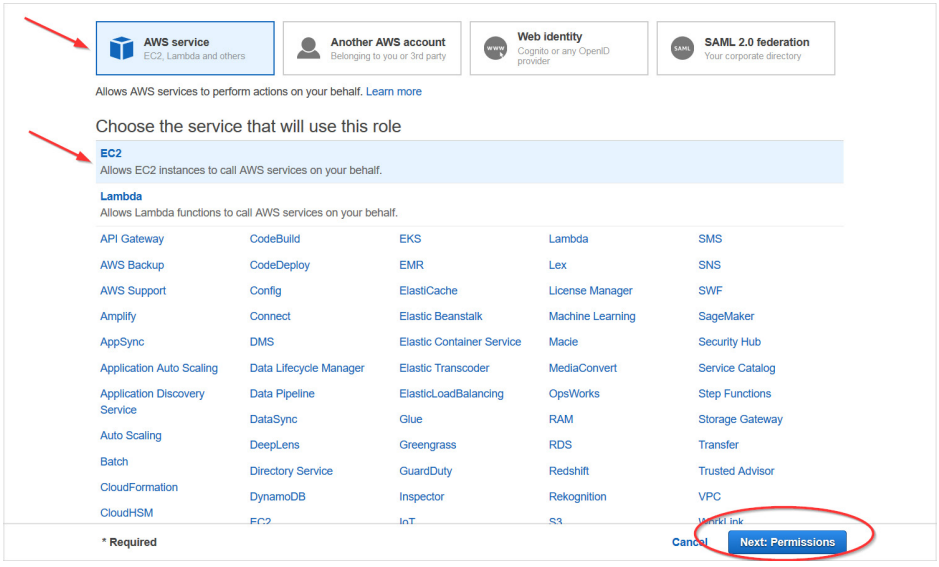
- You can use the wildcards, ***** and **?** within any ARN segment.
- An asterisk, *****, represents a combination of zero or more characters.
- A question mark, **?**, represents a single character.
- You can use multiple ***** and **?** in each segment, but a wildcard cannot span segments.

6. Copy the S3 bucket policy name to the Required Information section.

Create an IAM Role

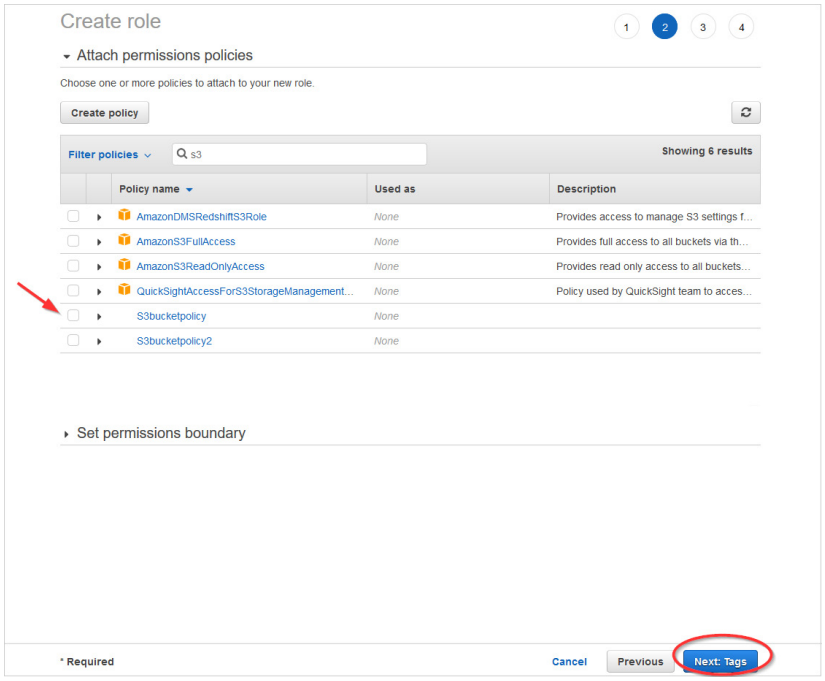
In this procedure, you will create an IAM role and attach it to your S3 bucket policy to access the bucket and its data.

1. From the IAM dashboard, select **Roles**. The Create Role page opens.
2. Click **Create role**.
3. Select **AWS service**, choose **EC2**, and click **Next: Permissions**.



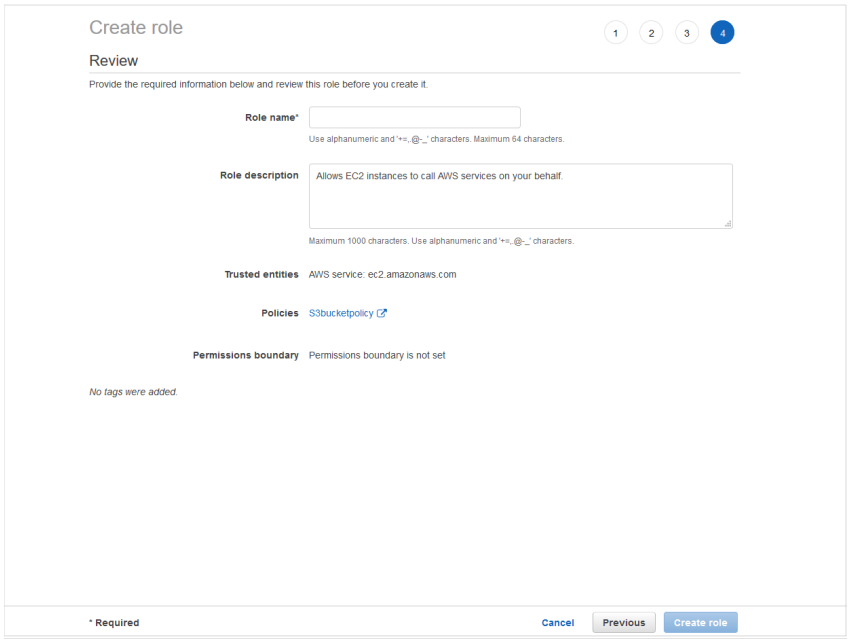
The Attach permissions policies page opens.

4. Select the checkbox next to the S3 bucket policy click **Next: Tags**.



5. Click **Next: Review**.

   The Create role page opens.



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

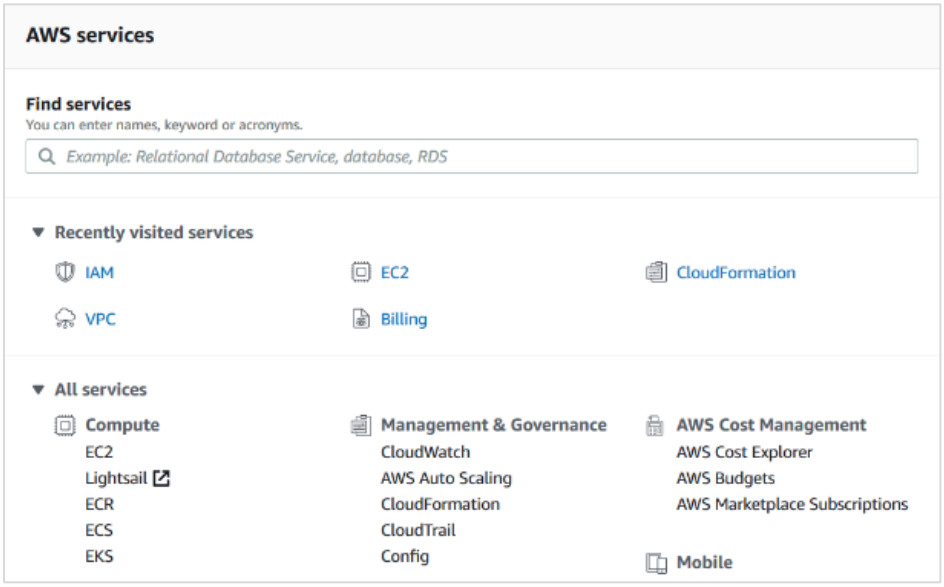
7. Copy the name of the role that will access to the S3 bucket to the Required Information section.

LAUNCH THE SELF-HOSTED AMI

CloudCheckr will let you know when the self-hosted **Amazon Machine Image (AMI)** is available in the AWS Management Console. The AMI contains all the information you need to launch your **EC2 instance**, which is the virtual server where you will run the self-hosted application.

1. Log in to the AWS Management Console.

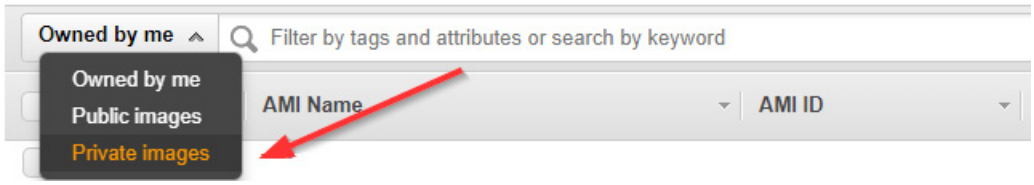
   The AWS services page opens.



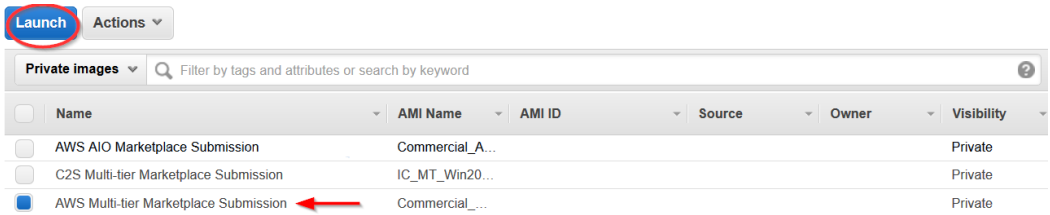
2. From the middle of the page, choose **Compute > EC2**.

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

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



5. Select the radio button next to the AMI and click **Launch**.



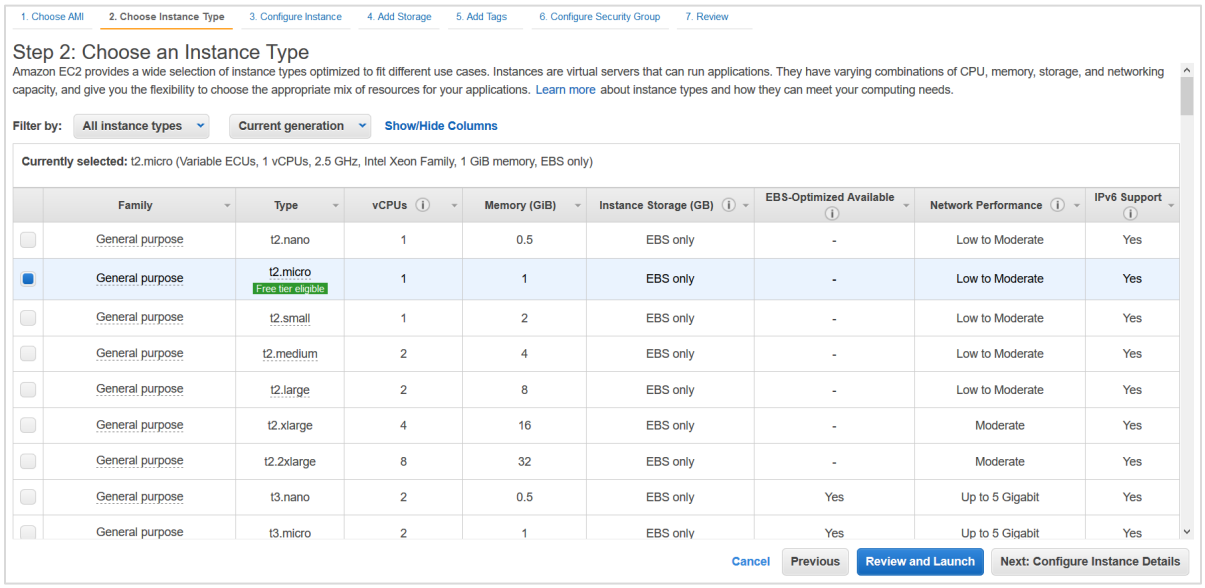
CONFIGURE THE EC2 INSTANCES

You need to configure three EC2 instances: one for your Web Console, one for the Scheduler, and one for the Workers.

Configure the Web Console

After you click **Launch**, AWS opens a wizard where you will configure your EC2 instances.

Since you already selected the AMI, AWS directs you to Step 2: **Choose an Instance Type**. This is where you choose the type of EC2 instance from where you will run the self-hosted application.

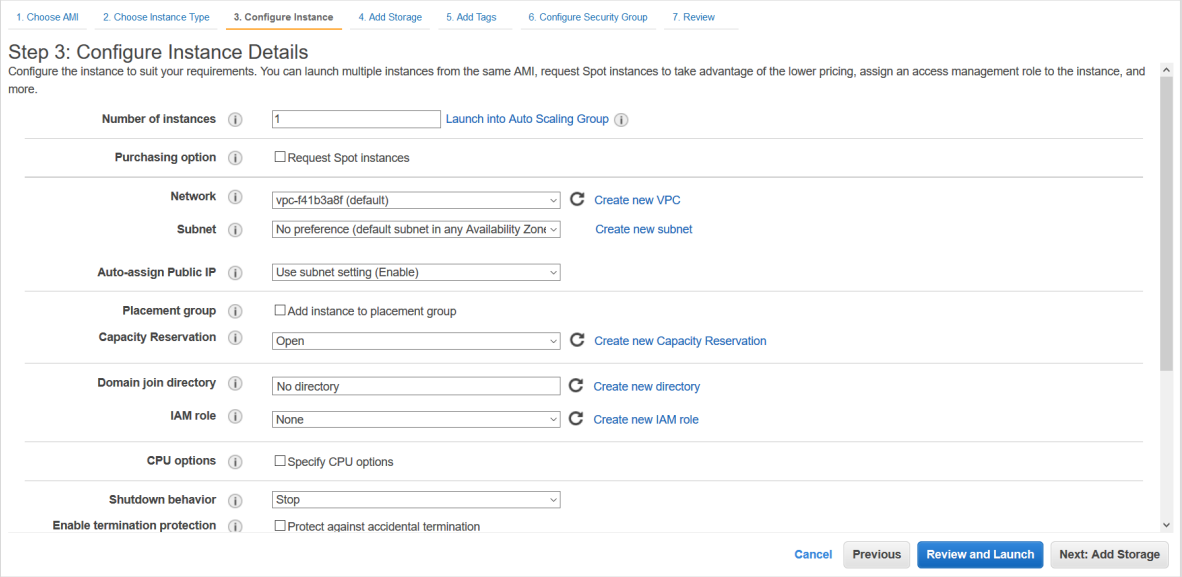


1. Select the checkbox next to **c5.large**.

   Note: CloudCheckr recommends **c5.large** as the best way to keep your costs down without sacrificing any performance. However, you should continue to monitor the cost and performance of your EC2 instance and modify the instance type for 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.



3. At a minimum, configure the following settings:

Option	Description	Action
Number of instances	The number of instances you want to deploy	Leave the default number of **1**
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	Choose a **public subnet** so you can access your EC2 Instance from the public internet
IAM role	Determines what you can and cannot access in an AWS account during a session	Select the IAM role that you created to have access to your S3 bucket

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 act as 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 purposes of this procedure, 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.

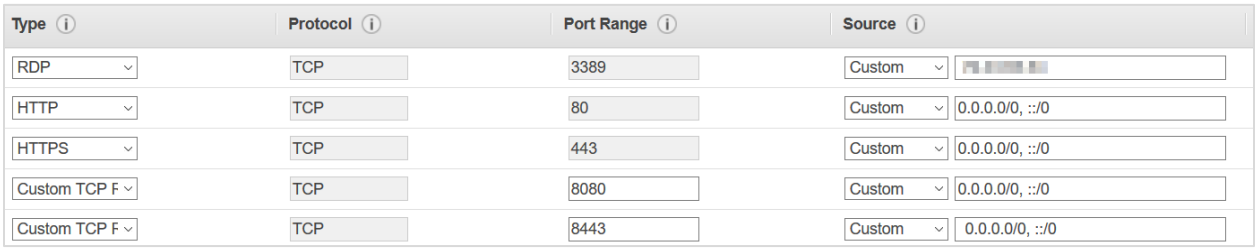
    b. Click **Add Rule** and create each of the following rules:

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 rules, keep the default 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
HTTP	TCP	80	Custom 0.0.0.0/0, ::/0
HTTPS	TCP	443	Custom 0.0.0.0/0, ::/0
Custom TCP F	TCP	8080	Custom 0.0.0.0/0, ::/0
Custom TCP F	TCP	8443	Custom 0.0.0.0/0, ::/0

To use an existing security group:

a. Select the **Select an existing security group** radio button.

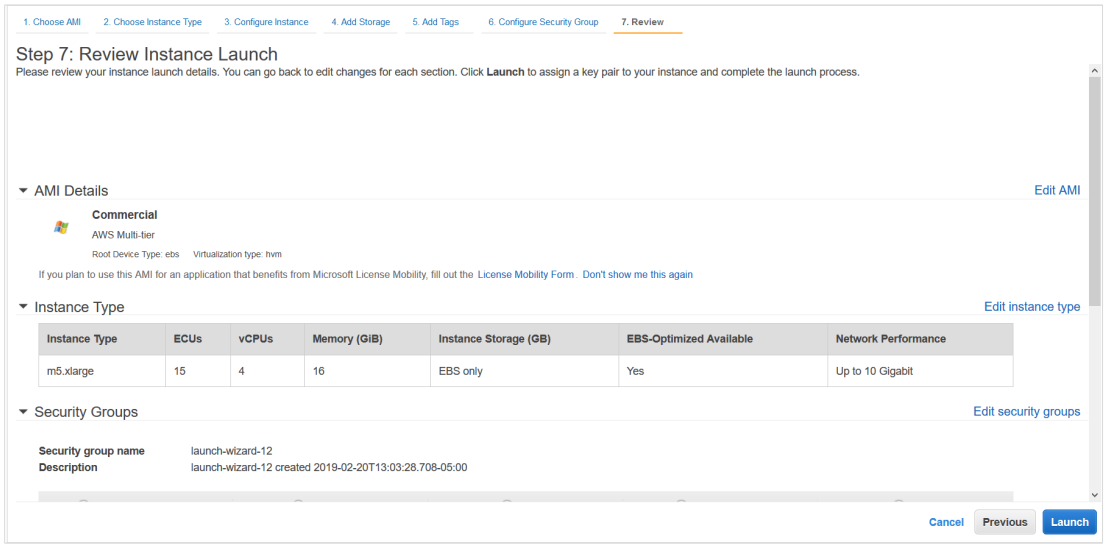
b. Select the checkbox(es) next to the security group you want to associate with your instance.

Select an **existing** security group

Security Group ID	Name	Description
sg-3474bb7c	default	default VPC security group
sg-078381c		ELB created security group
sg-034b5fd5		launch-wizard-1

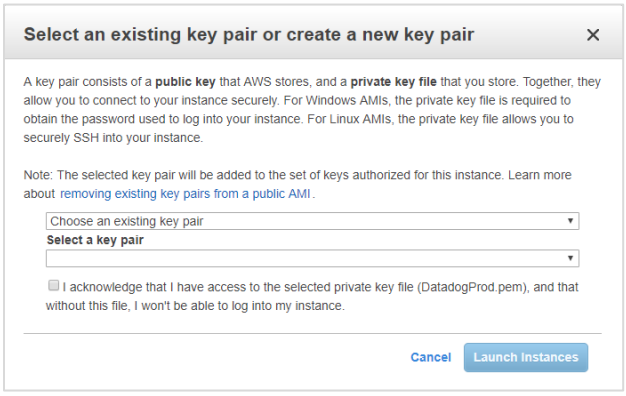
9. Once you have configured your security group(s), click **Review and Launch**.

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



10. Click **Launch**.

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



To choose an existing key pair:

a. Verify that **Choose an existing key pair** is selected in the top drop-down menu.

b. In the Select a key pair drop-down menu, select an existing key pair.

c. Select the **I acknowledge...** checkbox.

To create a new key pair:

a. In the top drop-down menu, select **Create a new key pair**. The Key pair name text box displays.

b. In the Key pair name text box, type the name of the key pair.

c. Click **Download Key Pair**. A .PEM file will download to your desktop.

d. Save the .PEM file because you will not be able to generate it again.

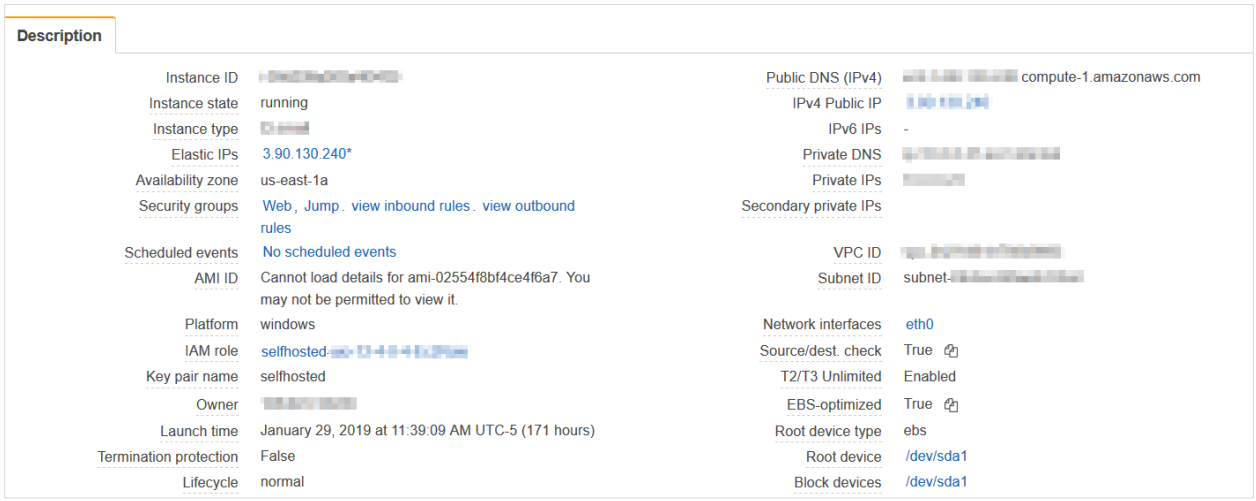
11. Click **Launch Instances**.

12. The Launch Status screen opens and will let you know when your instance is ready.

    Depending on the size and scale of your deployment, it should be ready in 5 to 10 minutes.

13. Once your instance is ready, return to the EC2 dashboard and select **Instances > Instances**.

14. Select the checkbox next to your EC2 instance to see details about your selected EC2 instance.



15. Copy the following values to the Required Information section:

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

Configure the Scheduler and Workers

To configure the Scheduler EC2 instance and the Workers EC2 instance, follow the instructions in the Configure the Web Console section **except** use **m5.large** for your instance type.

INSTALL THE SELF-HOSTED APP

This section shows you how to install a separate EC2 instance for the Web Console, Scheduler, and Workers.

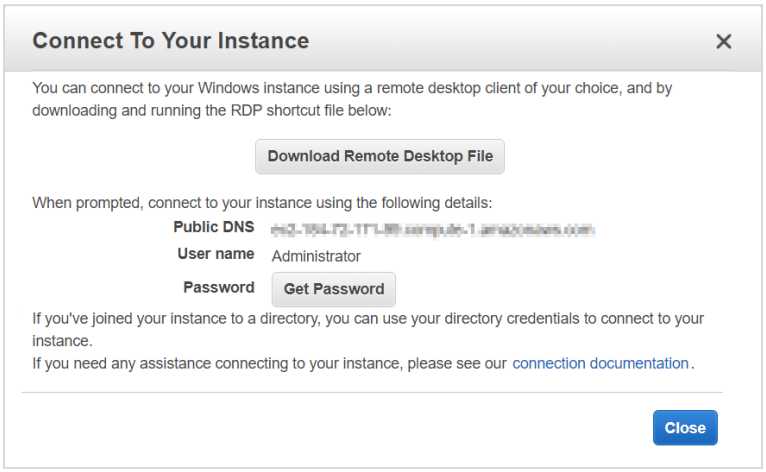
Although you **can** install the Scheduler and Workers on the same EC2 instance, we recommend that you install them on separate instances to accommodate for any increase in the size and scalability of your cloud deployment.

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.

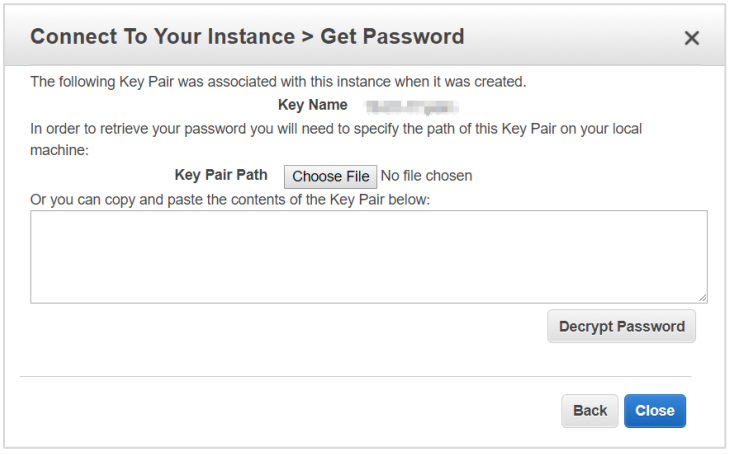
Install the Web Console

1. From your EC2 list in AWS, make sure that you selected your Web Console EC2 instance.

2. Right-click and select **Connect** from the fly-out menu. The Connect To... dialog box opens.



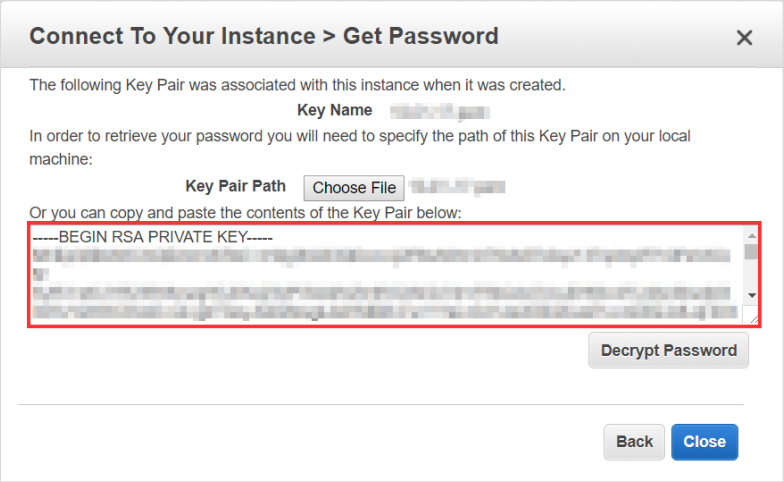
3. Click **Get Password**. The Connect Your Instance > Get Password dialog box opens.



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.

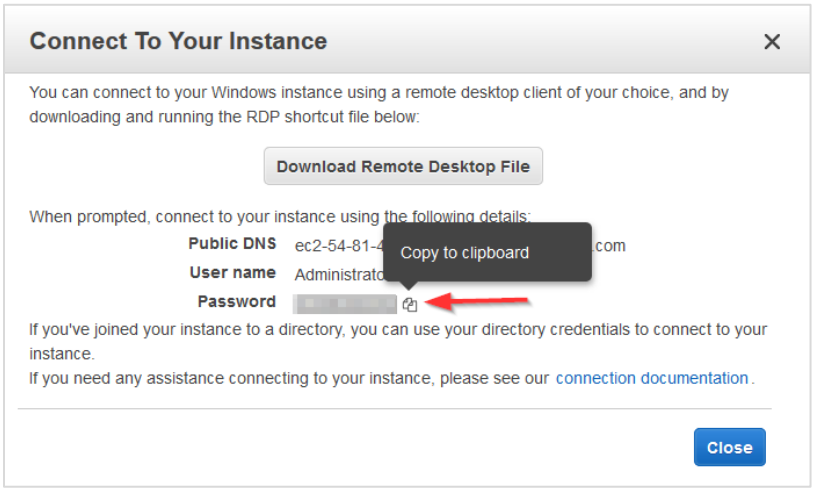


6. Click **Decrypt Password**.

   The default administrator password displays.

7. Hover to the right of the administrator password to display the **Copy to clipboard** icon.

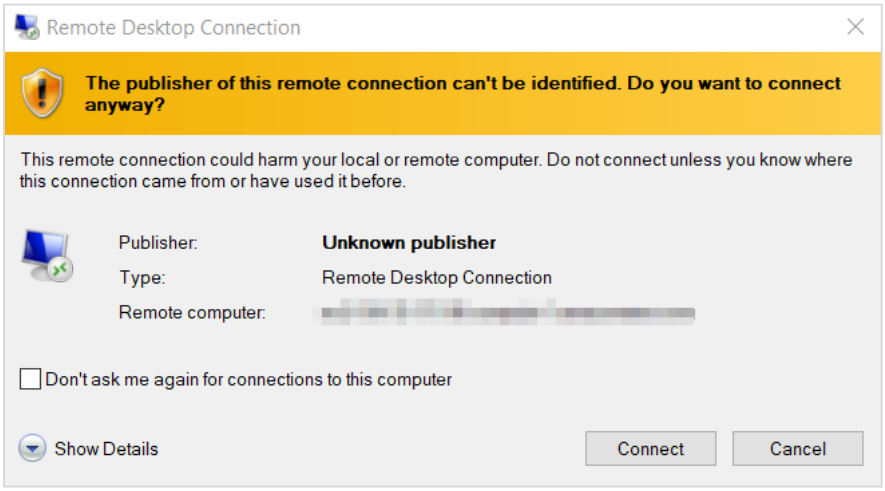
8. Click to save the password.



9. Click **Download Remote Desktop File**.

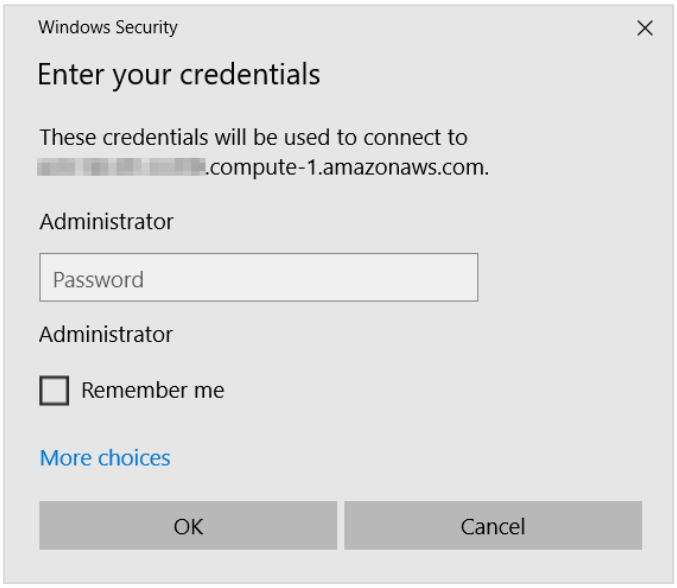
10. Open or save the .RDP file.

The Remote Desktop Connection dialog box opens.



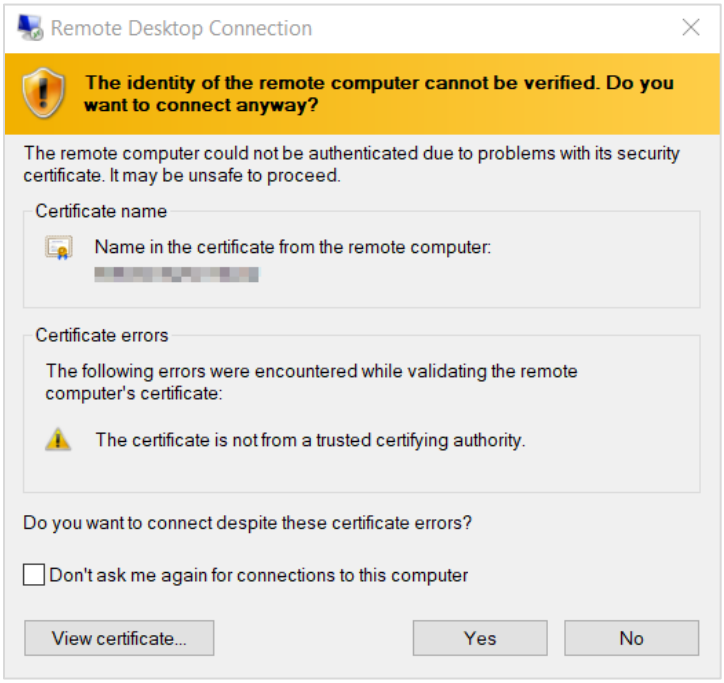
12. Click **Connect**.

    The next dialog box prompts you to provide your password.



13. 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.

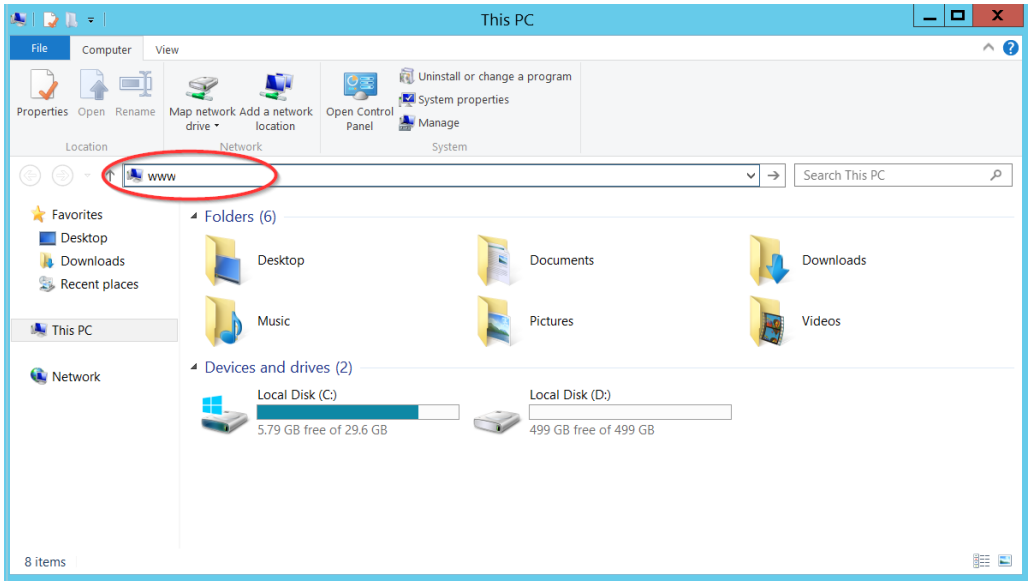


14. Click **Yes**.

    Your Remote Desktop session launches.

15. From the taskbar, click the **Folder** icon.

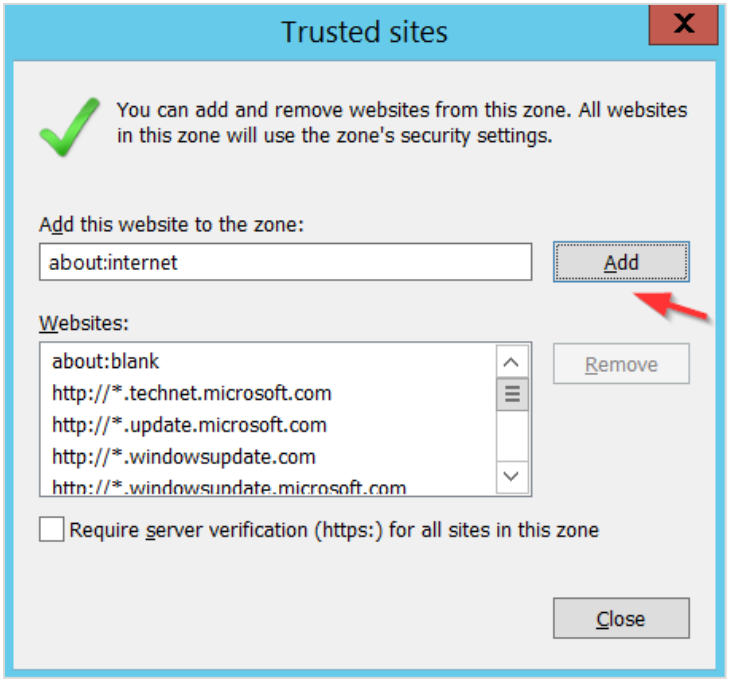
16. Type **www** in the search bar to open your browser and press **Enter**.



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

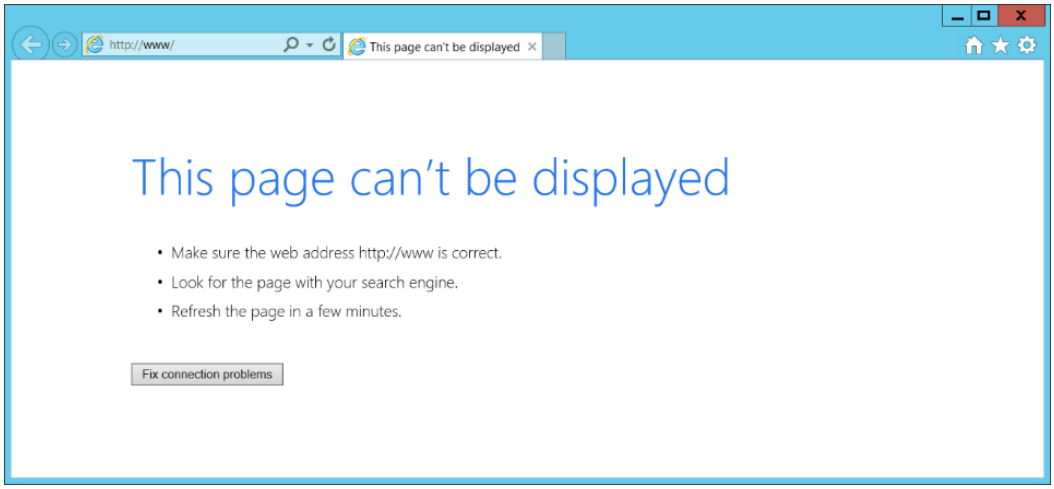
17. Click **Add.** The Trusted sites dialog box opens.

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



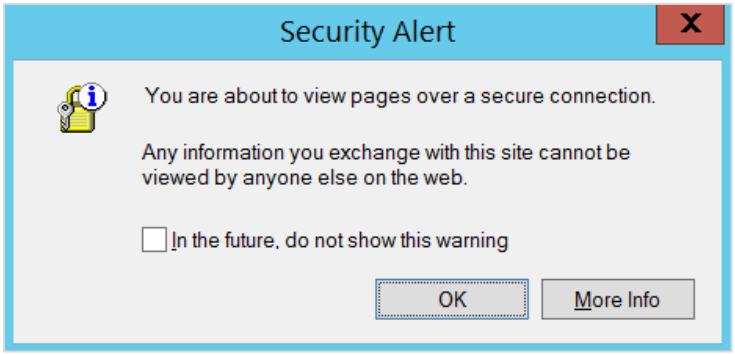
19. Click **Close**.

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



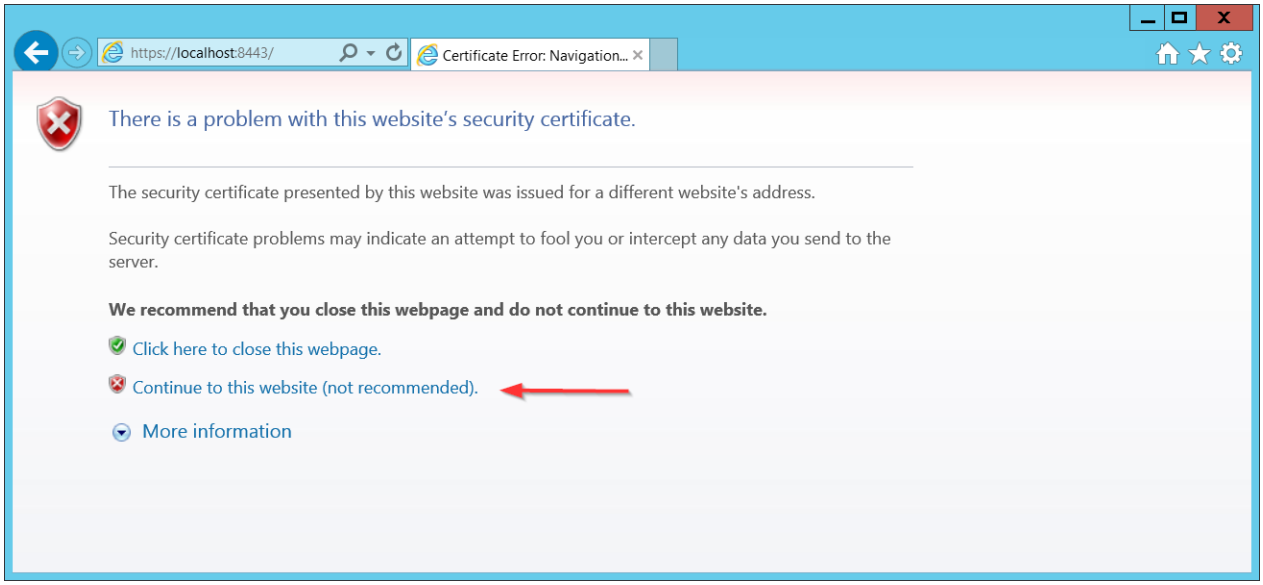
20. In the address bar, type **http://localhost:8080** and press **Enter**.

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



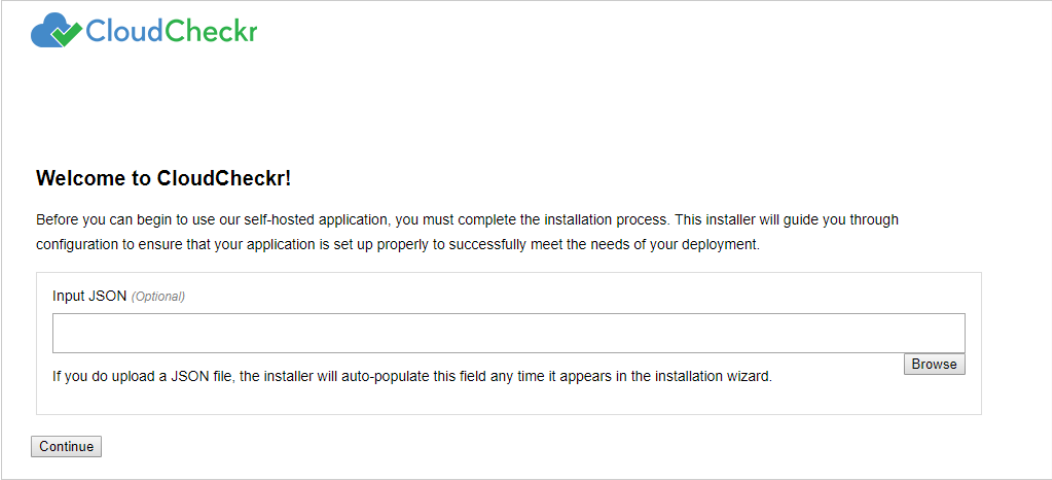
A warning about your security certificate displays.

22. Click **Continue to this website**.



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

The Web installer opens. The first screen is where you can upload a JSON file if you want the Web installer to auto-populate this field any time it appears in the installation wizard.

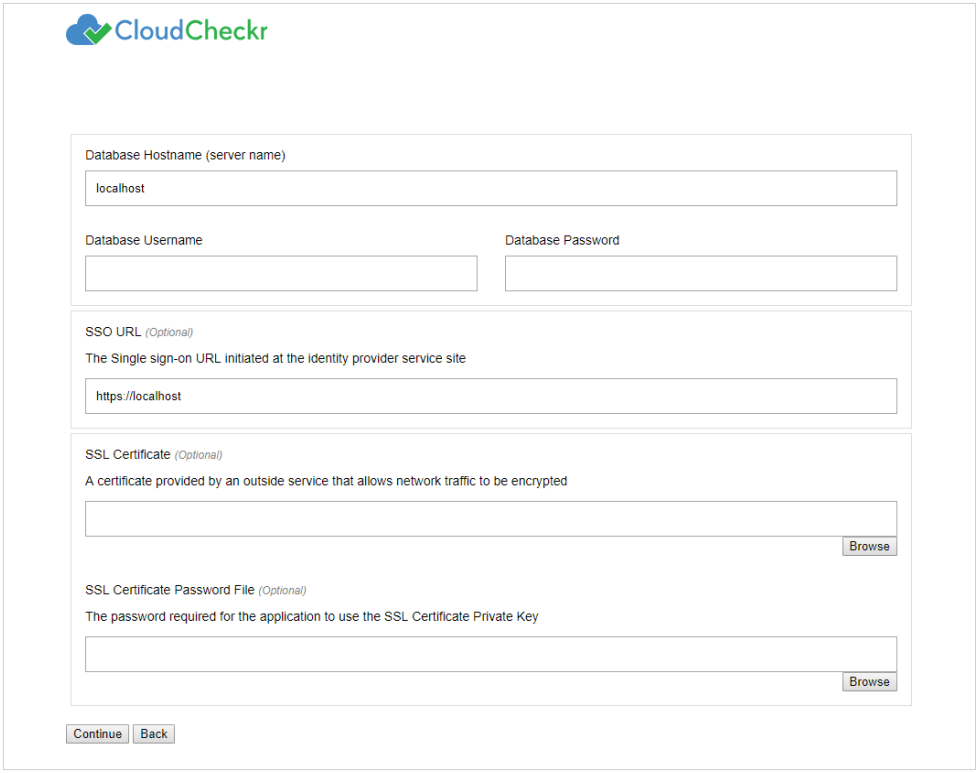


Note: The Input JSON text field is an **optional** feature. If you do not want to use the website to configure the self-hosted application, you can load the file using the command line:

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

24. Click **Browse** to navigate to the JSON file if you want to upload it. See the Input JSON File section.

25. Click **Continue**. The next screen is where you configure your security features.



26. Go to the Required Information section and copy the following values that you configured in the Create an RDS Database section:

    - **Database Hostname**: the endpoint of the RDS database
    - **Database Username**: the username that you use to connect to the RDS database
    - **Database Password**: the password that you use to connect to the RDS database

27. If applicable, provide the following information:

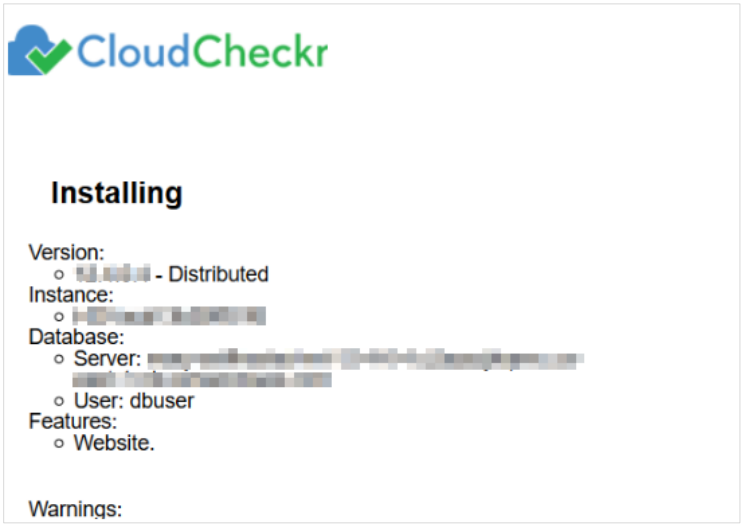
    - **SSO URL**: Single Sign-On URL
    - **SSL Certificate**: allows network traffic to be encrypted
    - **SSL Certificate Password File**: password required for the application to use the SSL certificate private key

28. Click **Continue**.

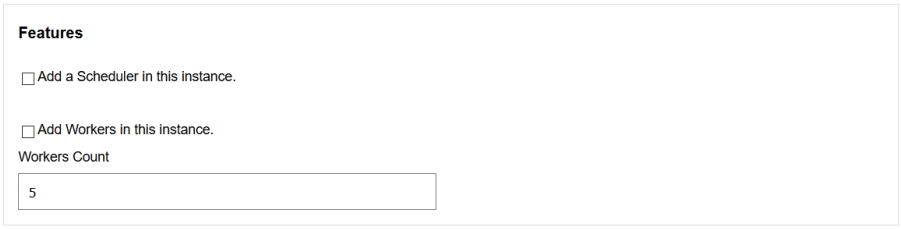
    The next screen in the Web installer opens. The first section in this screen:

    - identifies the version number of the self-hosted application
    - provides the EC2 Instance ID

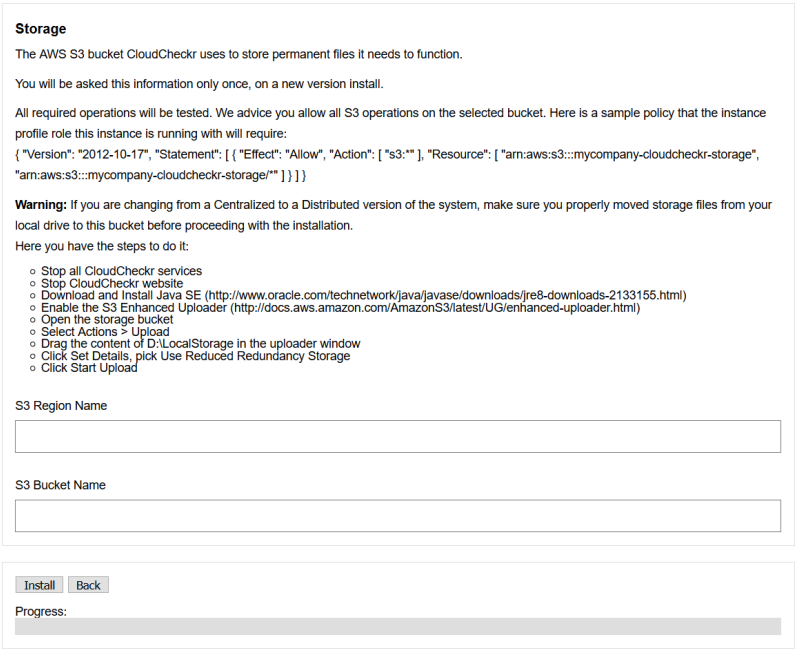
- verifies that the Microsoft SQL® server is available to communicate with the application
- identifies that the website (Web Console) is being installed



29. Scroll down to the Features section. Leave the default settings.



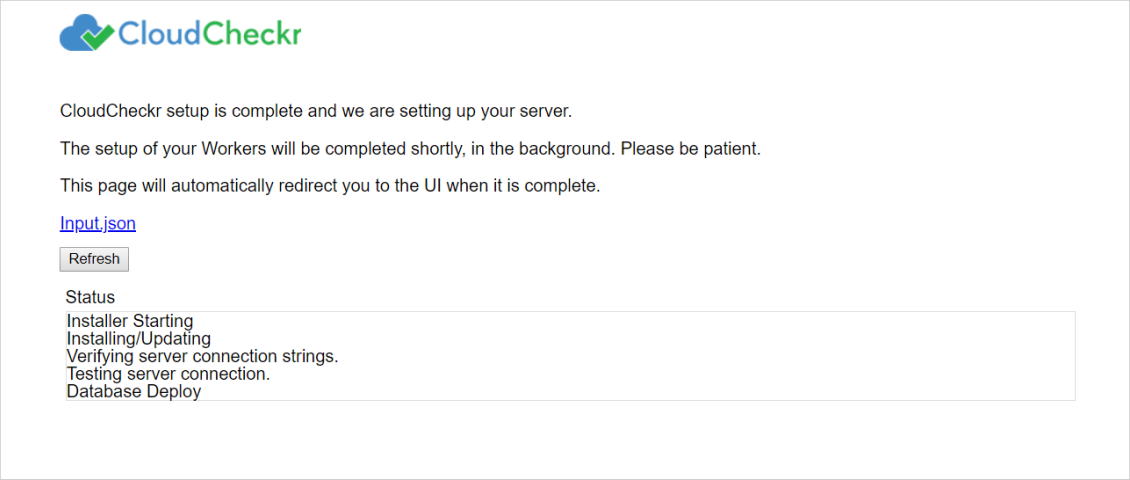
30. Scroll down to the Storage section. This is where you will identify the S3 region and bucket.



31. Copy the S3 region and bucket name from the Required Information section and paste them into the appropriate fields.

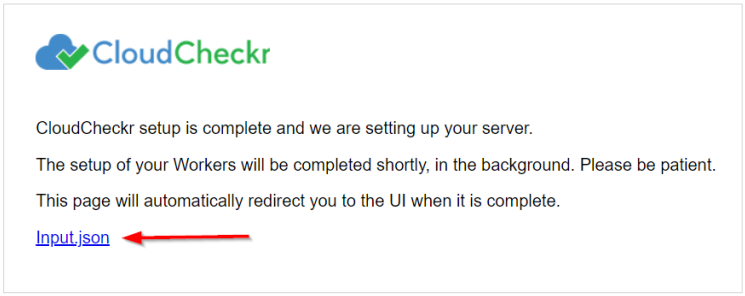
32. Click **Install**.

    The next screen provides status updates as the web installer completes its configuration task.



Input JSON File

Click the **Input.json** link to download the JSON file:

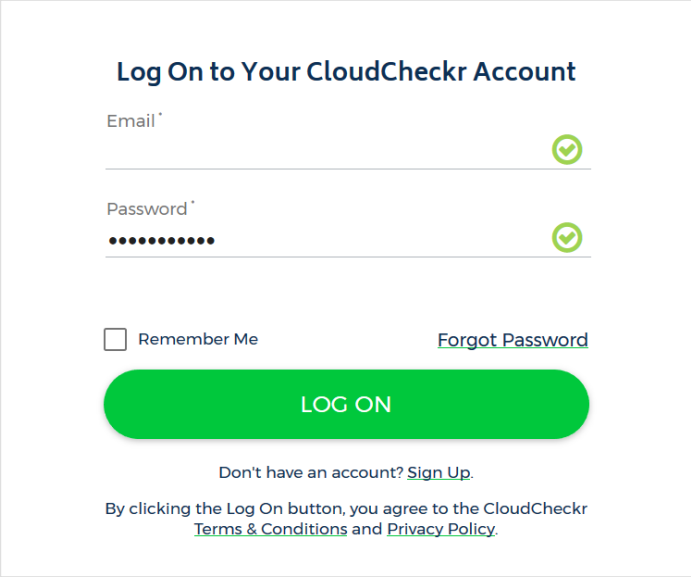


If you uploaded the file earlier, it will retain that same configuration. Since the filename is not important, you can rename the file to suit your needs. If you forget to click the link and you want to use the file later, go to:

C:\CloudCheckr\Input.JSON

Note: Installation may take a few minutes because the application must install the Microsoft Windows® services and deploy and populate the databases.

The log in screen of the application opens.



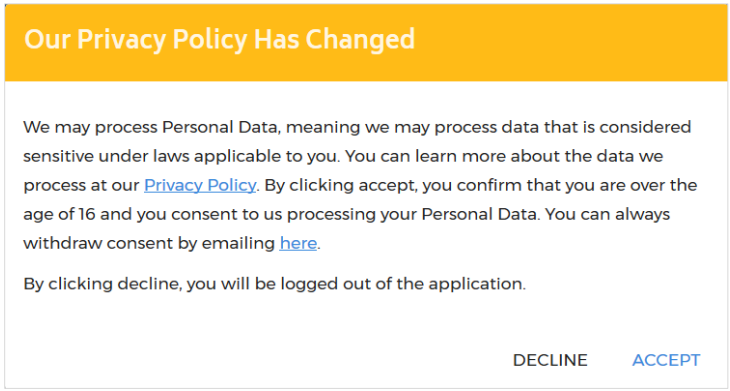
33. In the Email text field, type **sysuser**

34. In the Password text field, paste the **EC2 instance ID** of the Web console.

35. Click **LOG ON**.

    When you install the self-hosted application for the first time, you will see our privacy notice.

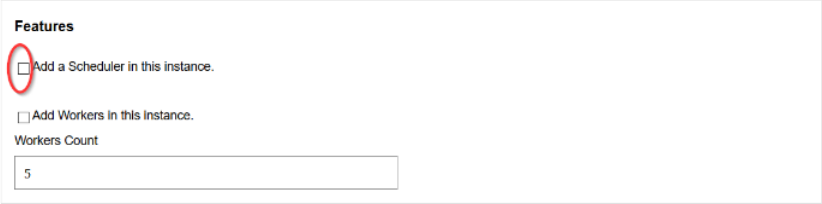
36. Click **ACCEPT** to acknowledge the changes to our privacy policy.



Install the Scheduler

Follow all the steps in the Install the Web Console section except:

- select the Scheduler EC2 instance from your EC2 list
- select the **Add a Scheduler in this instance** checkbox



Install the Workers

Follow all the steps in the Install the Web Console section except:

- select the Workers EC2 instance from your EC2 list
- select the Add Workers in this instance checkbox
- leave the Workers Count checkbox at the default number of **5**

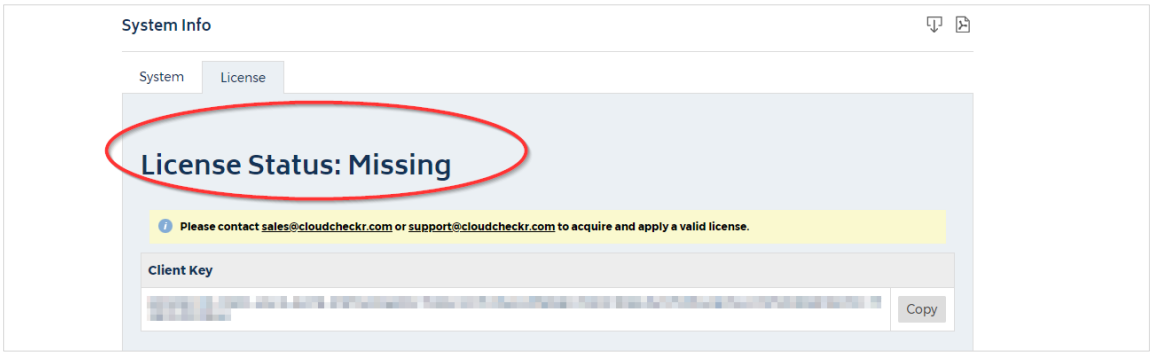
CONFIGURE THE SELF-HOSTED APP

License Your App

Note:

- If you purchased the AMI from the AWS Marketplace, you are **not required** to license the self-hosted application. Skip this section and go to the Create a Partner section.
- If you purchased the AMI privately from CloudCheckr, you are **required** to license the self-hosted application for all versions from the 12.4 release moving forward.

After you log in, you will see the System Info screen indicating that the license is **Missing**:



Before you can configure the self-hosted application, you must get a license file from sales.

1. In the License tab, go to the Client Key section, and click **Copy**.

2. Email your sales representative to request a new license. Be sure to include your client key.

   Note: If you don't have a sales representative or you can't reach your assigned representative, email the sales team. We prefer that you contact sales first rather than our support team so that we don't add to support's workload and overcomplicate the license process.

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

3. Once your sales representative provides you with a new license file, upload the license file:

    a. Save the license file, with an **LIC** extension, to a location in your self-hosted application.

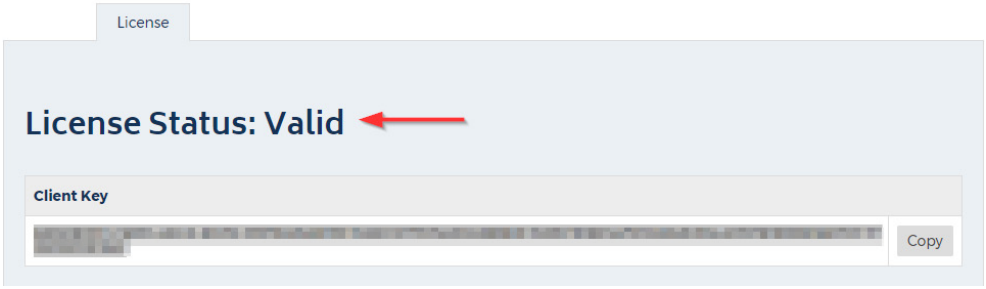
        - If you save the file to your local desktop, you won't be able to access the file when you remote into your self-hosted application.
        - The upload will fail if you don't save it with the correct extension.

    b. Navigate to the License tab on the System Info page.

    c. In the Update License section, click **Browse** to locate your license file.

    d. Click **Save**.

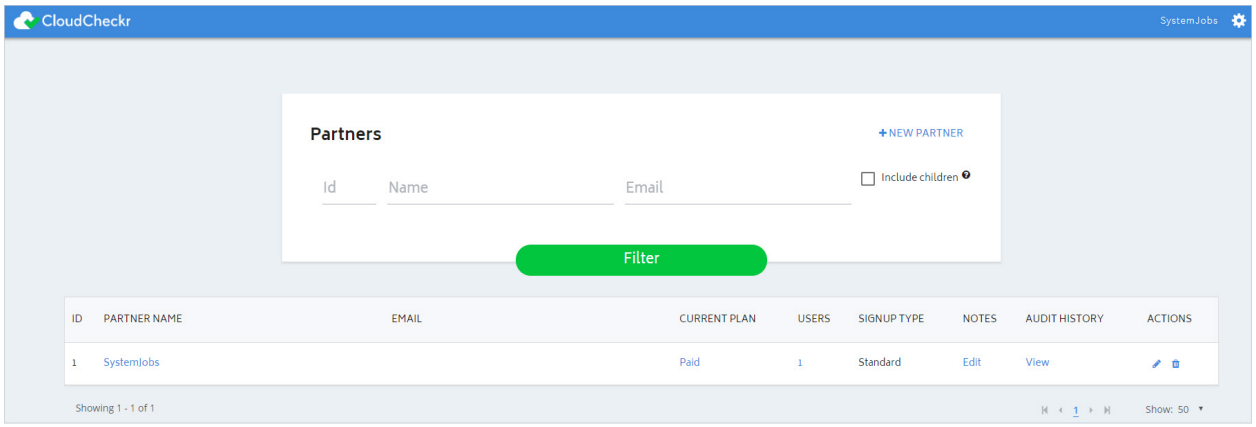
    Once the application loads the license file, the License Status changes to **Valid**.



Create a Partner

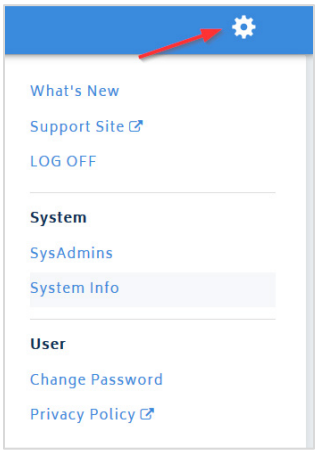
The first step in the configuration of your self-hosted application is to create a **partner**—the top-level container where you will store your accounts. More than likely, you will only need one partner—especially if you want all your accounts in one location.

The main landing page is the Partners page:



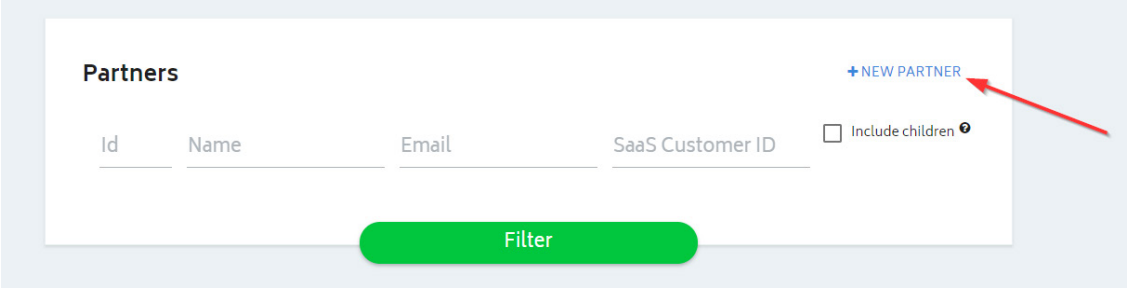
- If you just finished licensing your self-hosted application, click the **Back to Partners List** button in the License tab to return the Partners page.

Because you have not configured the application, you will only see the Settings icon is in the header bar. If you clicked **Settings** at this point, you would only have access to basic functions like viewing your system and license information and changing your password:

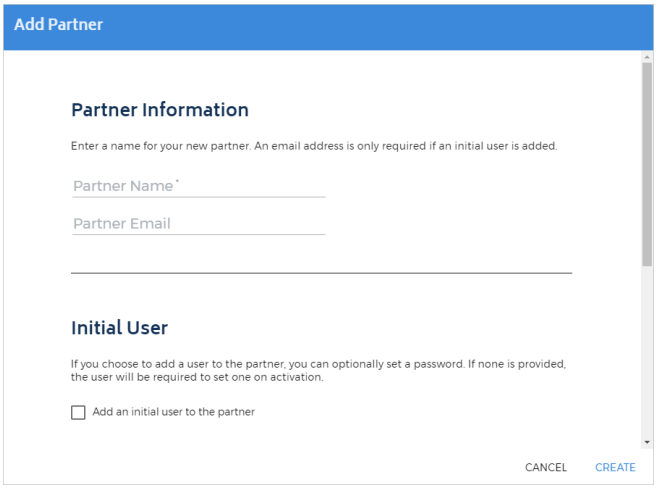


After you create your partner, more functionality will become available to you.

1. Click **+ NEW PARTNER**.



The Add Partner dialog box opens.



2. In the Partner Name text field, type a partner name.

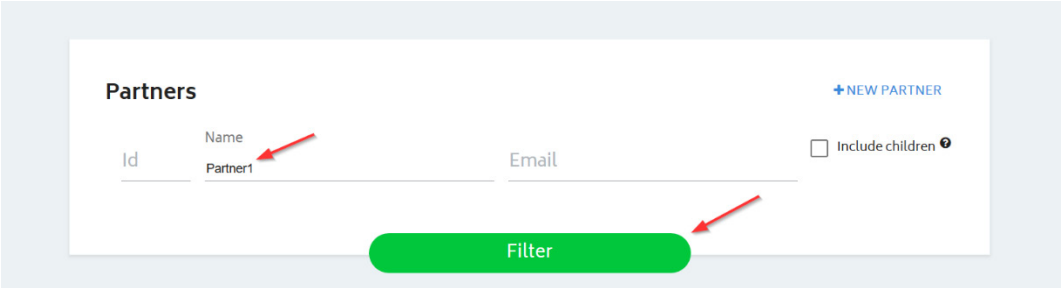
3. In the Partner Email text field, type an email address.

4. Click **Create**.

   A message indicates that the application added the partner successfully.

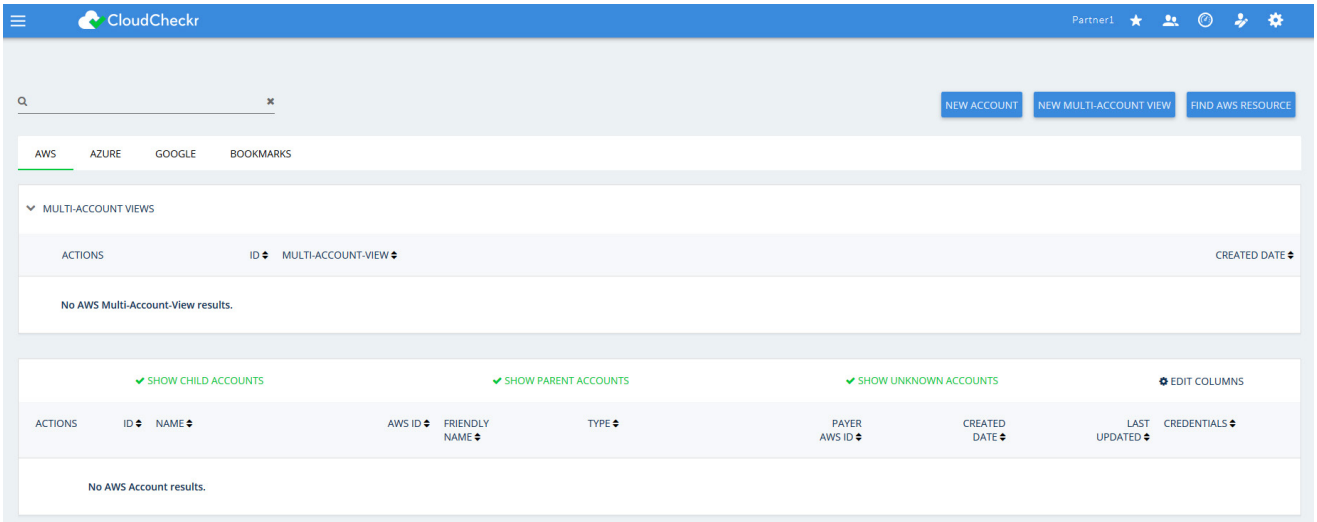
5. Click **OK**.

6. Type the name of your new partner and click **Filter**.



CloudCheckr adds your new partner to the partner list.

7. Click the **partner name** to open the Accounts page. This is where you will add the accounts you want to associate with your partner.



Because you configured a partner, you now have access to more functions in the header bar:

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

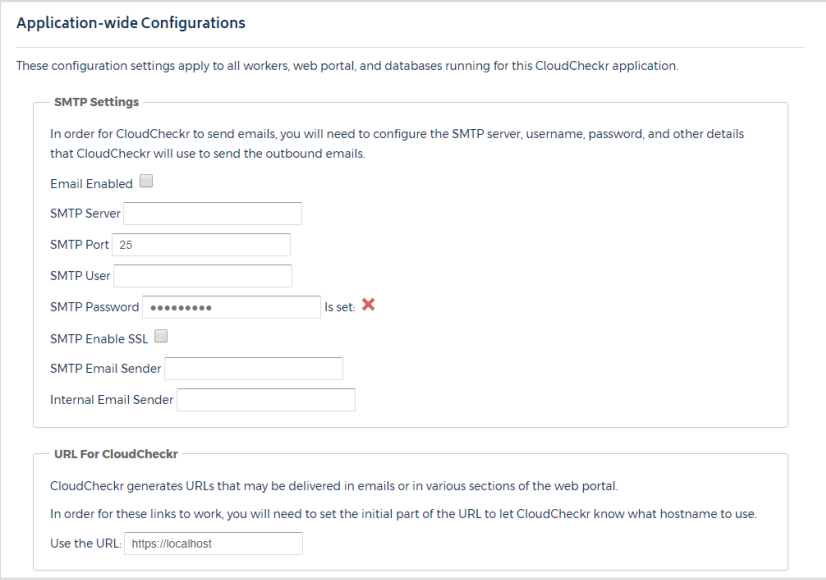
Before you can create your account(s), you need to finish some back-end configuration steps.

Complete the Back-End System Configuration

Out of the box, the self-hosted application does not have the same functionality as CloudCheckr's 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.

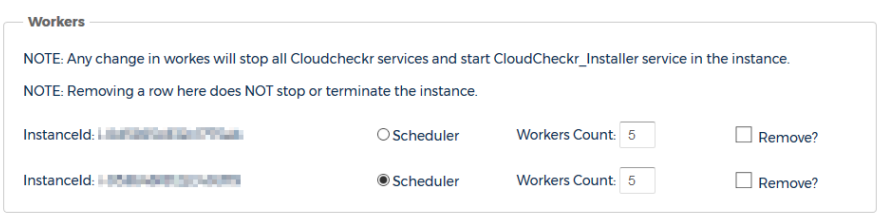


2. Scroll down to the SMTP section and configure the settings that will allow the self-hosted application to send emails on new user activations, alerts, and report data.

3. Scroll down to the URL For CloudCheckr section and provide the URL that you want to display on any system-generated emails.

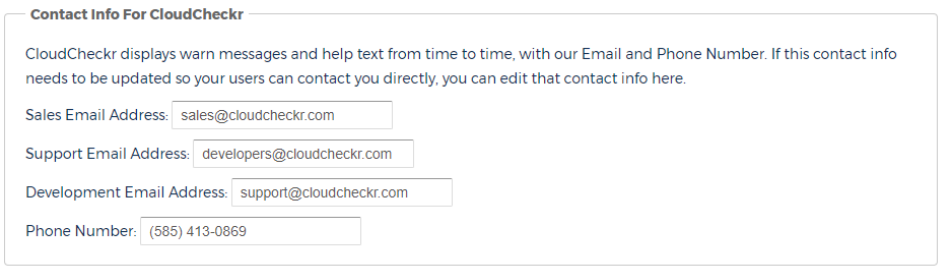
   Note: The default **localhost** will display the DNS for the EC2 instance that is hosting your self-application. This URL is external-facing so you can use it to send emails.

4. Scroll down to the Workers section to see the default number of workers.

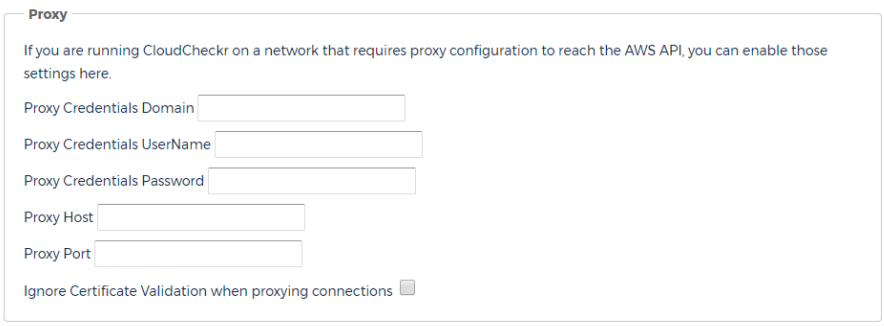


   Note: If you change and save the workers count, CloudCheckr will re-install the workers to match the new counts. This will stop data from being processed and may render the UI unresponsive for a few minutes. You can use the application once the screen becomes responsive again.

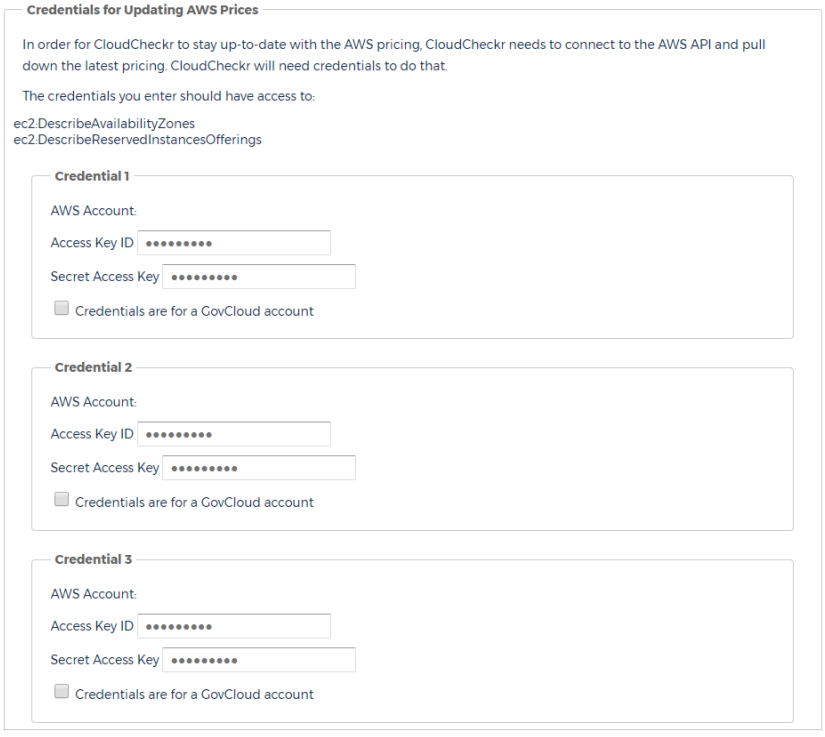
5. Scroll down to the Contact Info for CloudCheckr section and change the default email addresses and phone number if you want your users to contact you directly.



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



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



Create a Trusted User

When you assume a role, AWS gives you temporary security credentials to access other AWS accounts. This functionality is referred to as a **cross-account role**. To create a cross-account role, you must first create a **Trusted User**—an IAM user whose credentials enable the cross-account role to work with the self-hosted application.

Follow these instructions to create a trusted user:

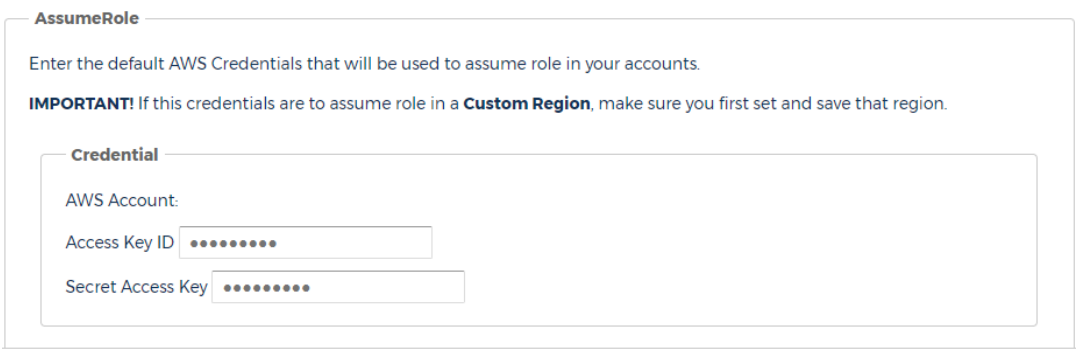
1. Copy this Trusted User policy and replace **AWS ACCOUNT ID** with your 12-digit AWS account ID.

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

2. Create the Trusted User policy using the instructions in the Create an IAM Policy section.

3. Create a trusted user and attach it to the Trusted User policy using the instructions in the Create an IAM User section.

4. Paste the access and secret keys of the Trusted User into the AssumeRole section.



5. Click **Save Settings** to save all the configuration changes you made to the self-hosted application.

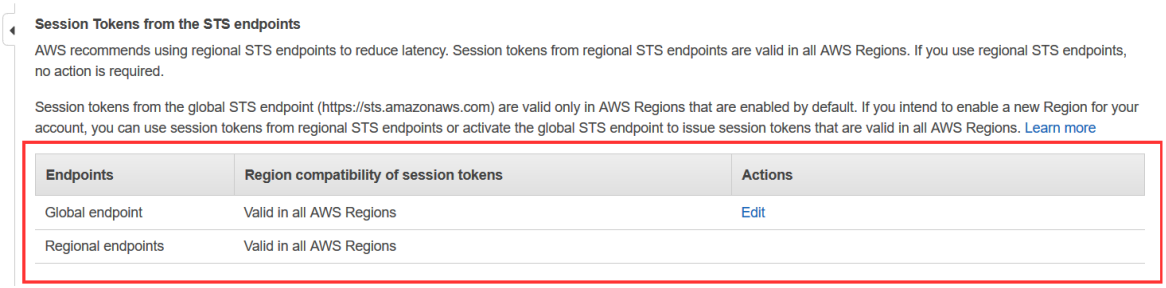
6. Copy the name of the Trusted User Policy and Trusted User to the Required Information section.

Create Trusted User Not in a Standard Region

If you need to credential an account that is **not** in a standard AWS region, such as Hong Kong, you must complete these additional steps when configuring your trusted user:

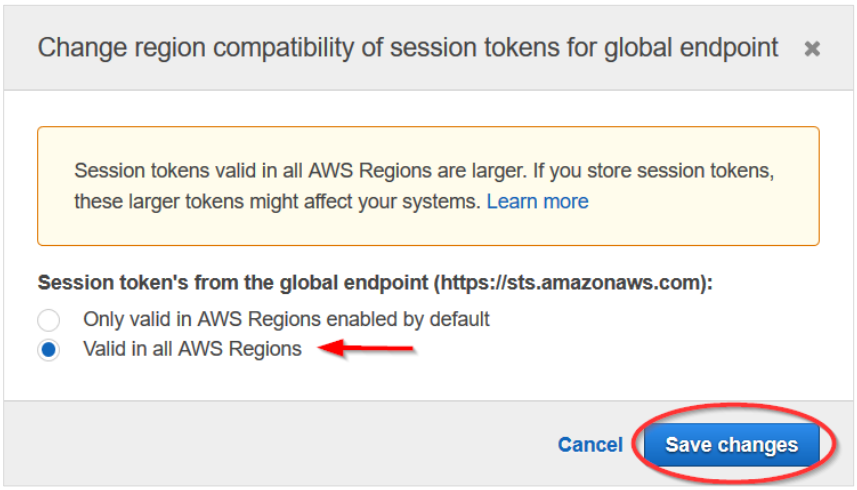
1. From the IAM dashboard, click **Account settings**.

   The middle of the right pane now displays a section on global and regional endpoints.



2. In the Global endpoint row, go the Actions column and click **Edit**.

3. In the dialog box, select the **Valid in all AWS Regions** and click **Save Changes**.



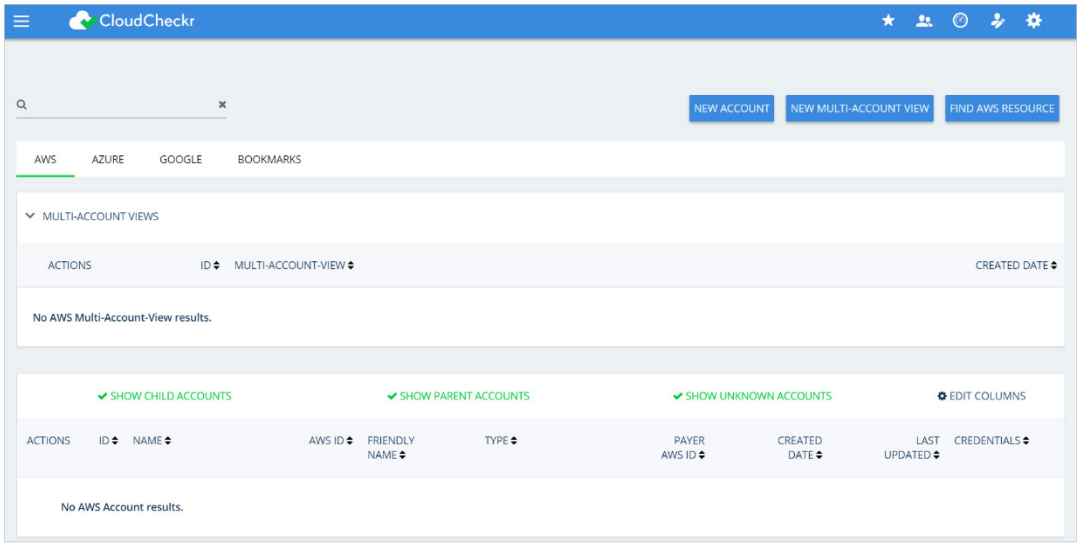
Create an Account

Now that you have configured all the back-end settings, you can create an account or accounts. 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**.

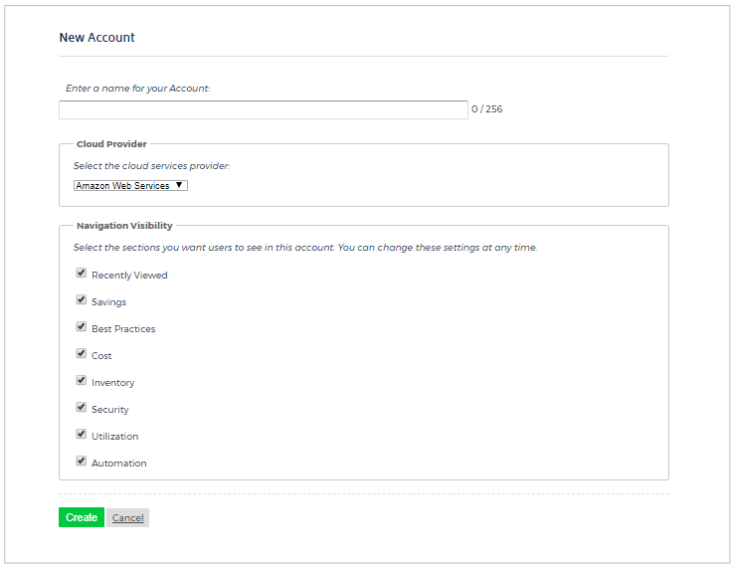
   Note: You can also go to the Partners landing page and click **your partner name**.

   The Accounts page for your partner displays.



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

   The New Account screen displays.



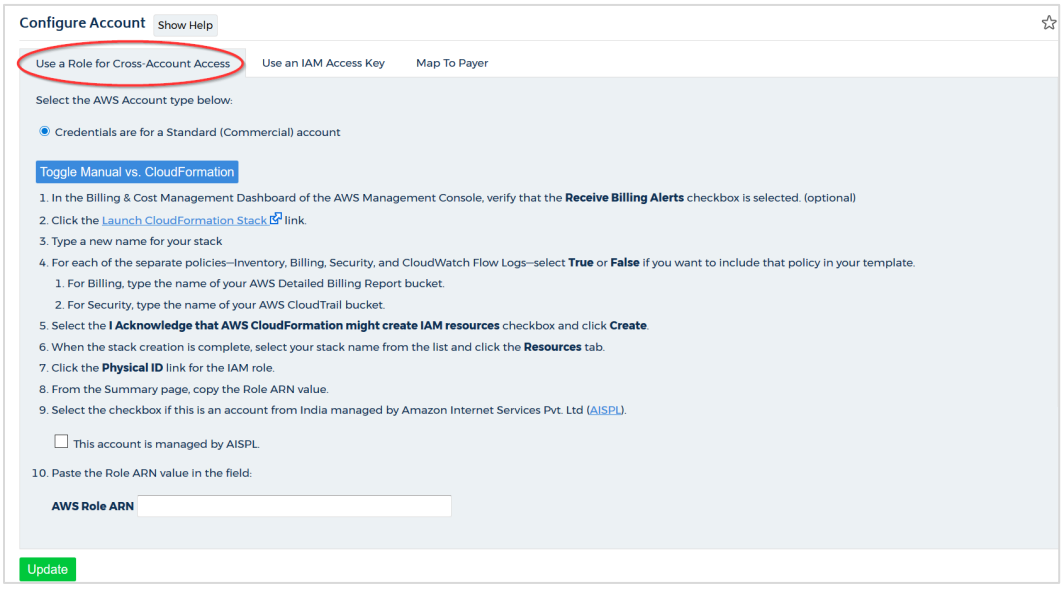
3. Type a unique name for your account and in the Cloud Provider section, select **Amazon Web Services**.

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**: lists more than 550 recommendations based on industry compliance standards
   - **Cost**: includes reports on daily spend, Reserved Instances (RIs), access to billing data, and more
   - **Inventory**: contains Summary, Detail, and Trending reports on your cloud provider's offerings
   - **Security**: helps you audit, conduct forensics, and manage other security issues
   - **Utilization**: provides metrics, visualization, analysis, and right-sizing recommendations
   - **Automation**: helps automate administrative tasks related to security and maintenance

5. At the bottom of the New Account page, click **Create**.

   The Configure Account page opens, and the Use a Role for Cross-Account Access tab is visible because you added a trusted user.



Create Least Privilege Policies

As part of your configuration, you must create **least privilege policies**, which are documents you will attach to your cross-account role that enable CloudCheckr to access the AWS data it needs to create its reports. Each least privilege policy provides permissions to a core function in our application:

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

1. Using the instructions in the Create an IAM Policy section, create the least privilege policies based on the documents found in the Appendix: IAM Policies.

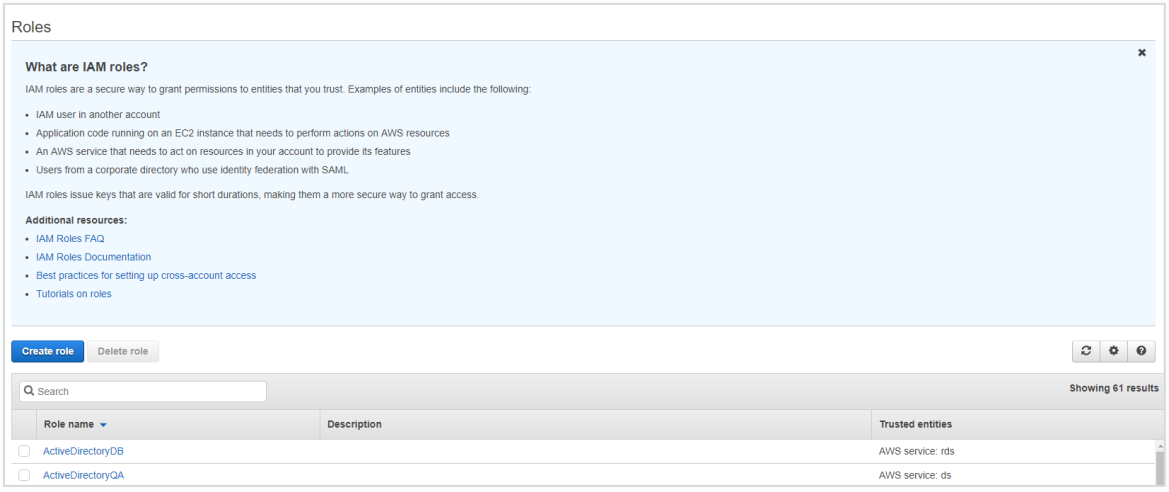
2. Copy the names of the least privilege policies to the Required Information section.

Create a Cross-Account Role

To finish your account configuration, you must create a cross-account role in AWS and apply those credentials in CloudCheckr.

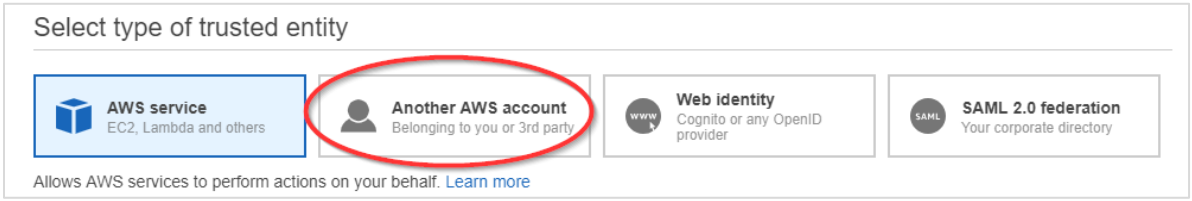
1. In the AWS Management Console, scroll down to the Security, Identity & Compliance section and select **IAM**.

2. From the dashboard, click **Roles**. The Roles page opens.

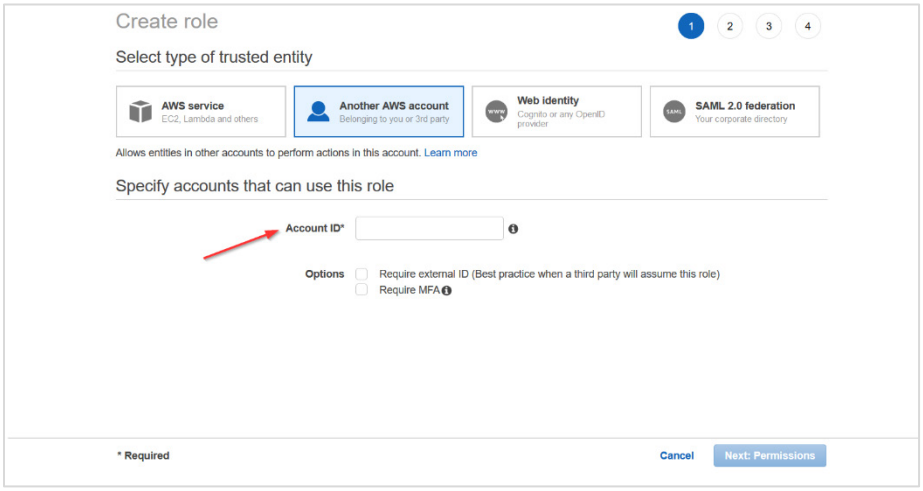


3. From the middle of the page, click **Create role**.

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



The screen prompts you to add an Account ID value.

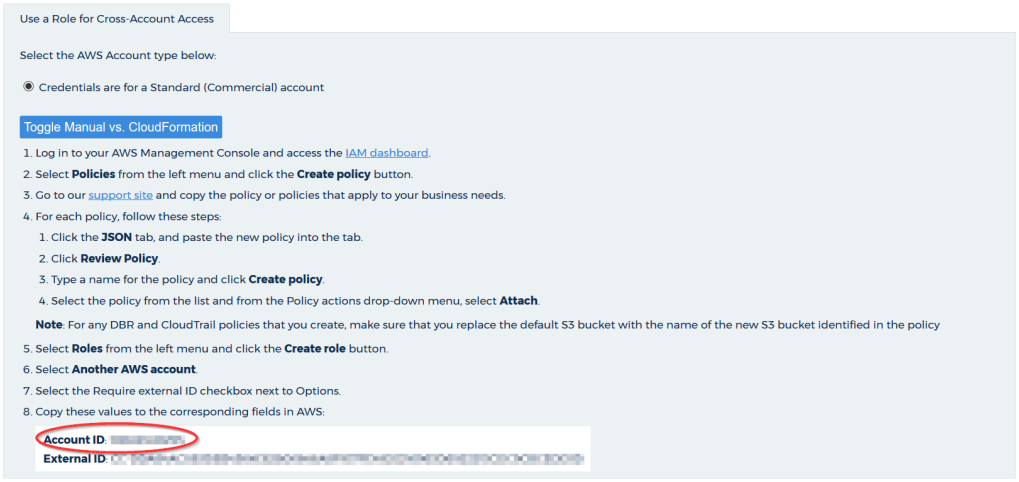


5. Get the account ID from the self-hosted application:

    a. Return to the Configure Accounts page.

    b. Click **Toggle Manual vs CloudFormation** to see the manual cross-account instructions.

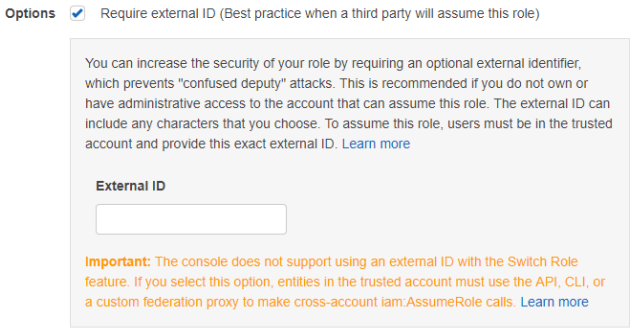
    c. Copy the Account ID.



6. Return to the AWS Management Console and perform the following steps:

    a. Paste the Account ID.

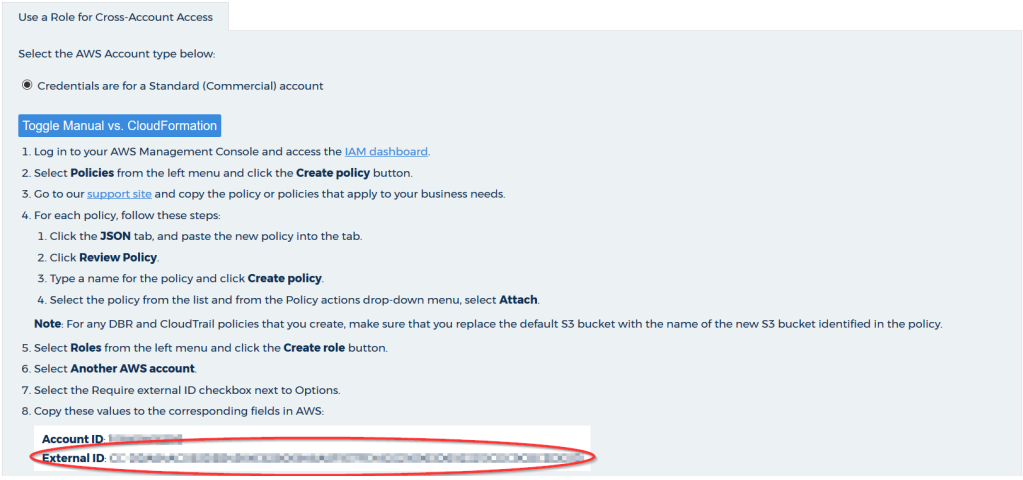
    b. In the Options section, select the **Require external ID** checkbox.



7. Get the External ID value from the self-hosted application:

    a. Return to the Configure Accounts page.

    b. Copy the external ID identified in the instructions.



8. Return to the AWS Management Console and paste the external ID value.

9. Verify that the Require MFA radio button is not selected.

10. Click **Next: Permissions**.

11. Select the checkbox next to each least privilege policy and click **Next: Tags**.

12. Click **Next: Review**.

13. The Review page opens.

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

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

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

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

17. 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** to complete the configuration of your cross-account role.

18. Copy the name of the cross-account role to the Required Information section.

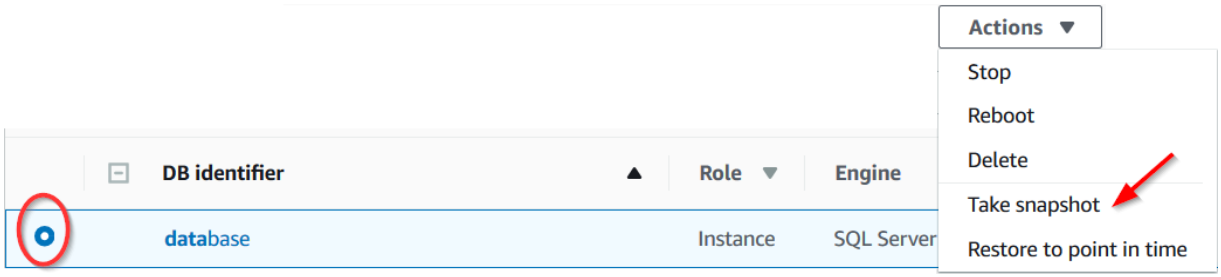
UPGRADE THE SELF-HOSTED APP

In this procedure, we show you how to upgrade the self-hosted application:

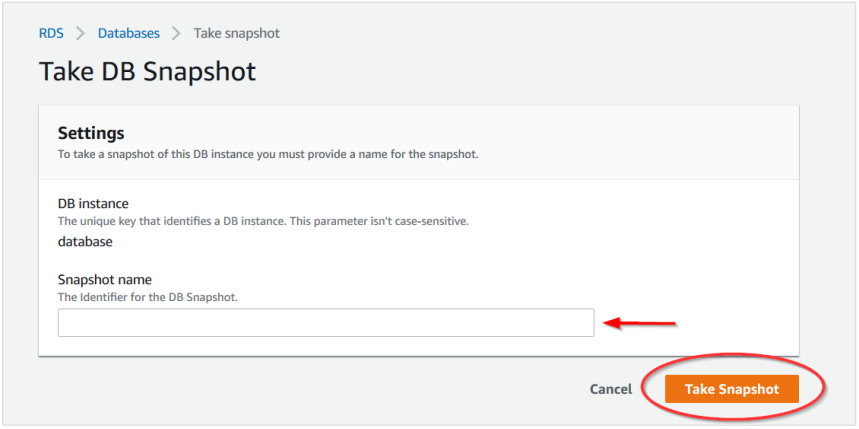
- create an RDS snapshot in AWS
- launch the new AMI
- install the new application
- license the new application

AWS stores your data in an RDS database. Before you upgrade the self-hosted application, you need to create a snapshot of that RDS database as a backup.

1. Return to the AWS Services page.
2. From the Database section, select **RDS**. The Amazon RDS page opens.
3. From the dashboard, click **Databases**.
4. Select your database from the list, and from the Actions menu, select **Take snapshot**.



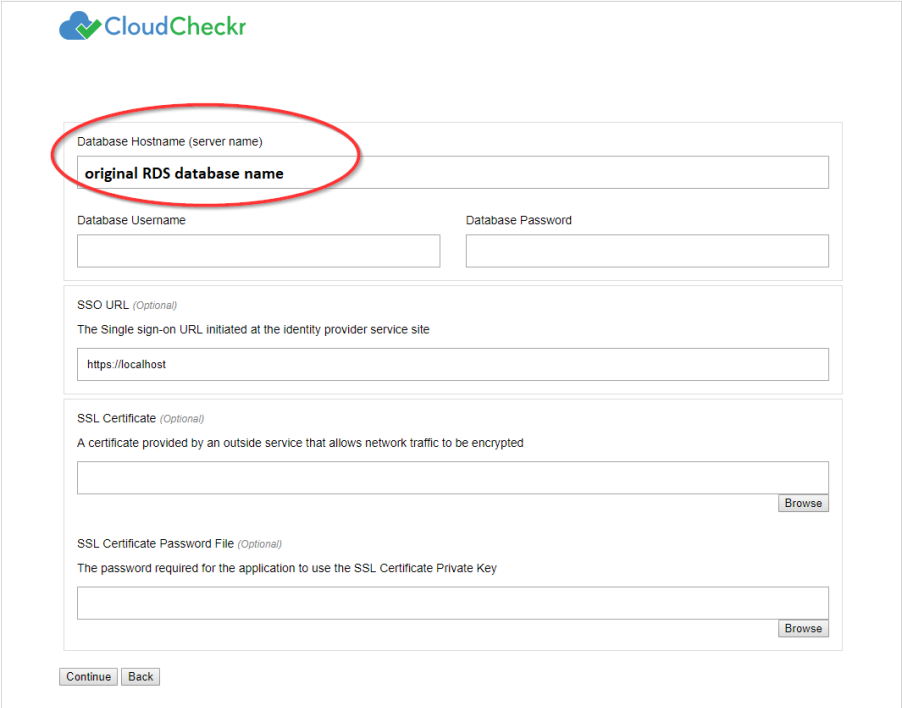
5. In the Take DB Snapshot dialog box, type a name for your snapshot and click **Snapshot**.



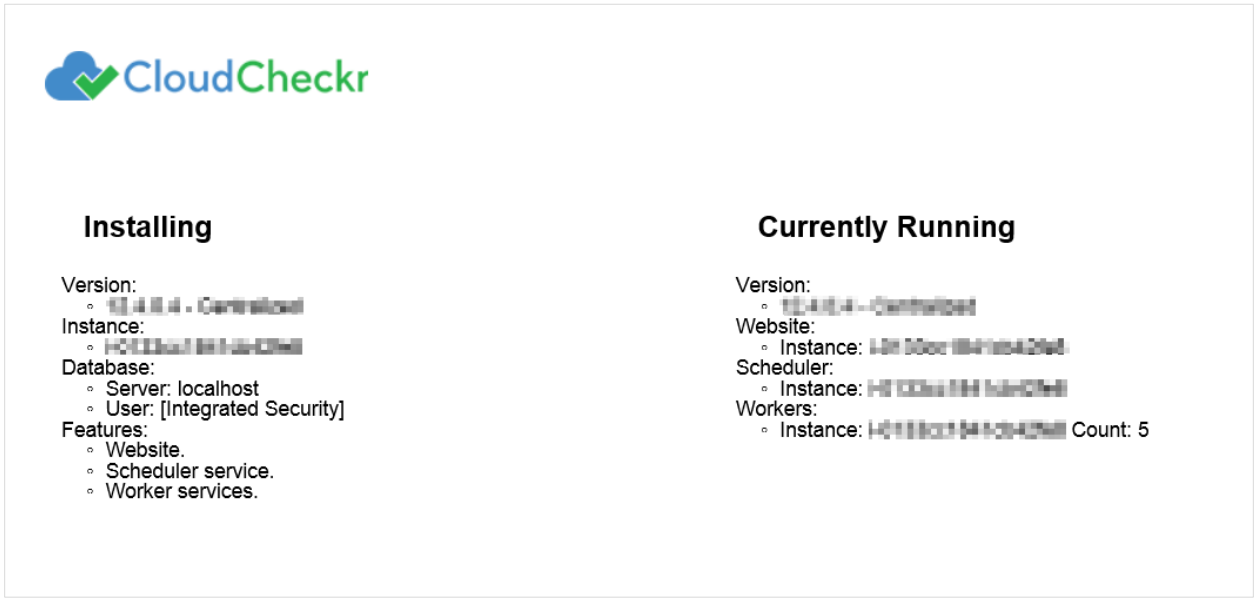
6. Complete the Launch the Self-Hosted AMI section making sure to select the **latest** AMII.

7. Complete the steps in the Configure the EC2 Instances section.

8. Complete the steps in the Install the Self-Hosted App section—making sure to type the name of your original RDS database when you get to the Database Hostname screen:



Note: During installation, the web installer will show you the version you are installing and the version you are currently running:



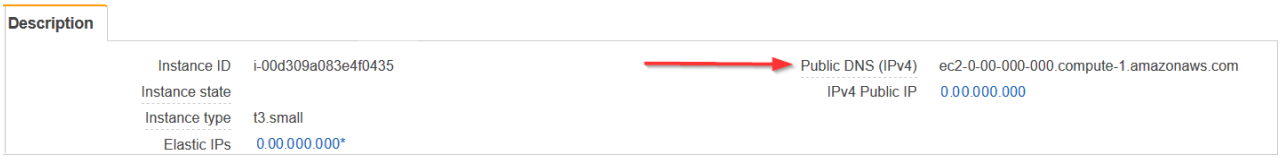
9. If you purchased a private offering, follow the License the App section to upload the new license file.

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.

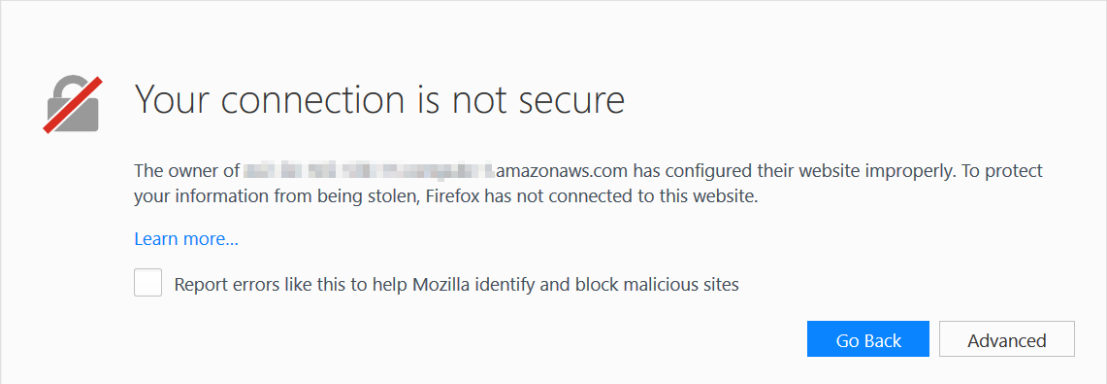
The external public hostname resolves to the public IP address or the Elastic IP address, which allows your instance to communicate to the internet.



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.

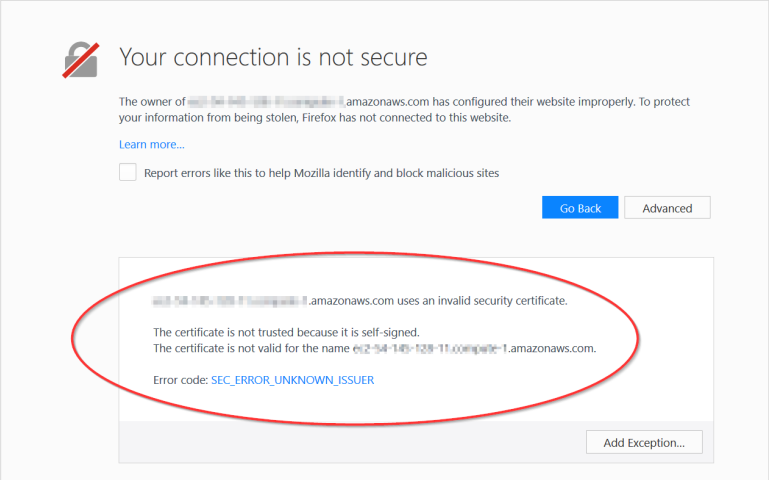
When the configuration is complete, a warning message indicates that your connection is not secure.



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

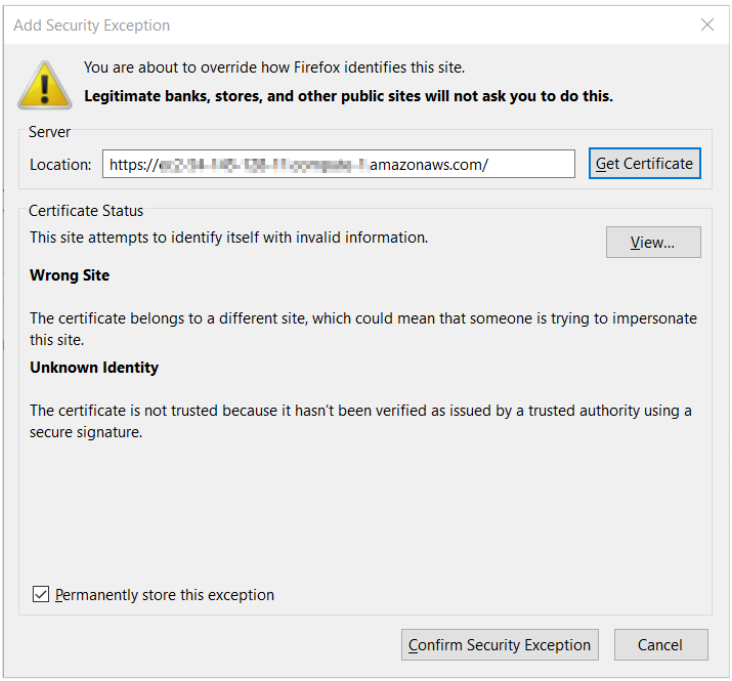
The application requires a secure connection with a certificate owned by the domain. Since you are launching the application in a self-hosted environment, it cannot automatically create a certificate.

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.

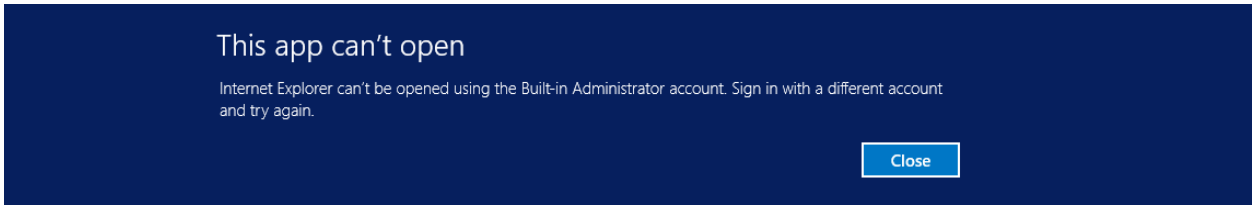
The Add Security Exception dialog box opens.



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?

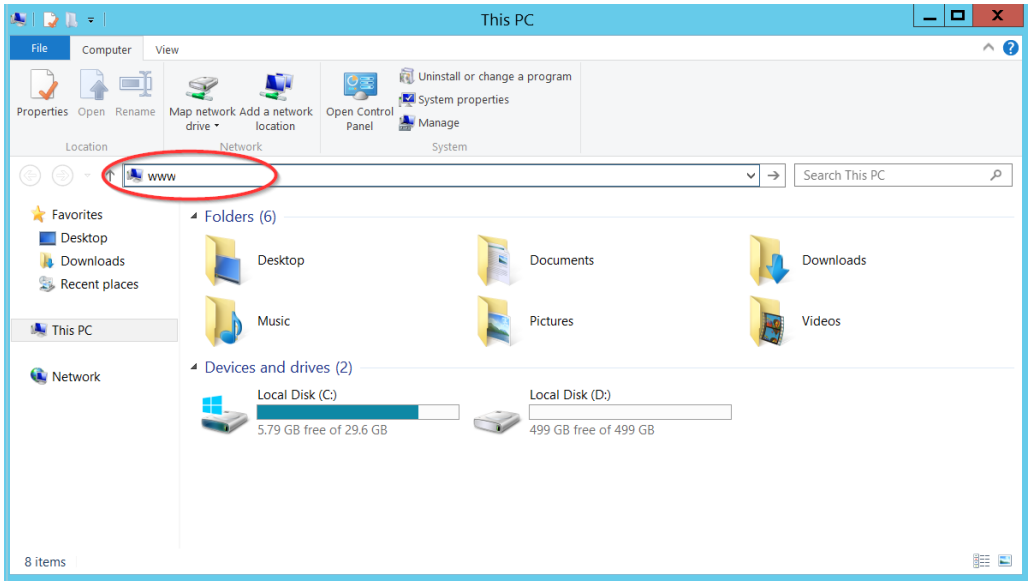
Your remote desktop session runs in a Microsoft Windows® 2012 R2 server environment, which is not compatible with newer applications like Internet Explorer. As a result, since you are logging in as an administrator, you will get an error message when you select **Start > Internet Explorer**:



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.

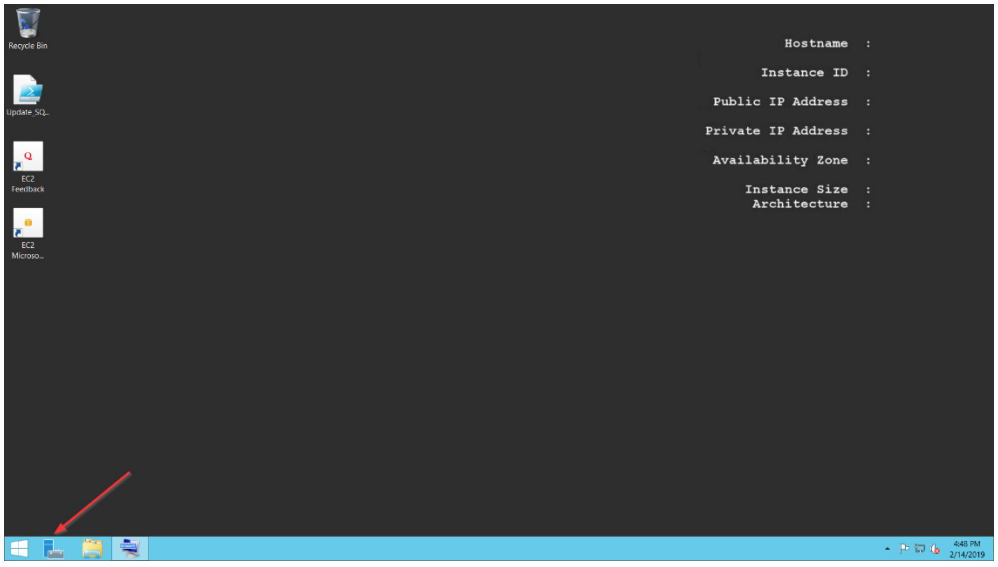


3. Follow the remaining steps in the Install the Self-Hosted App section to complete your connection to your EC2 instance.

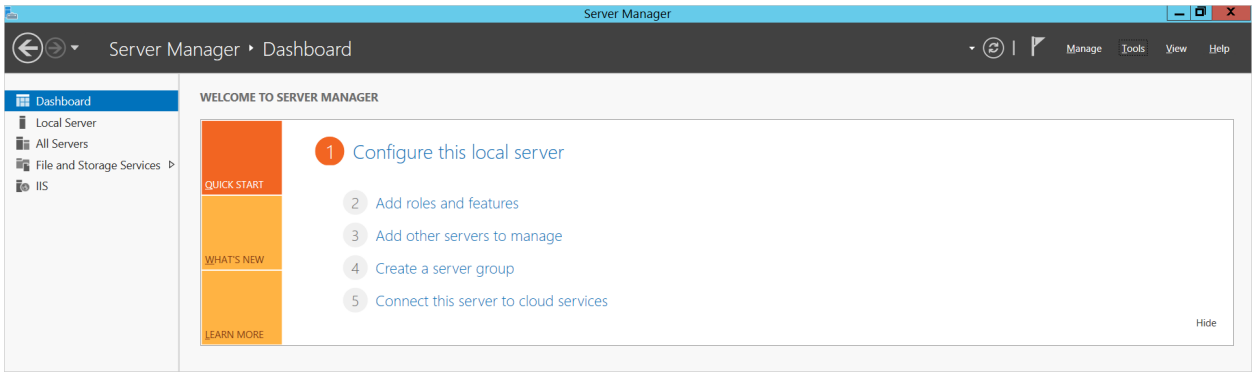
Where Is My D: Drive?

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.

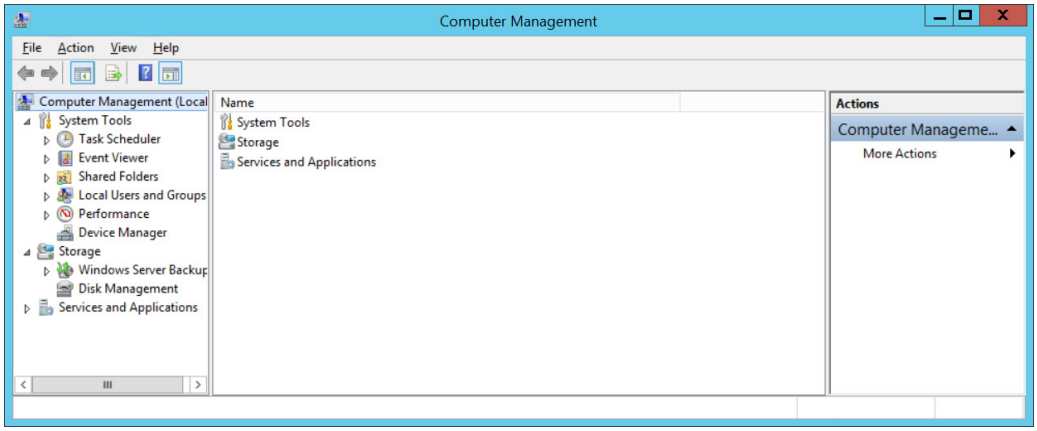


The Server Manager Dashboard opens.

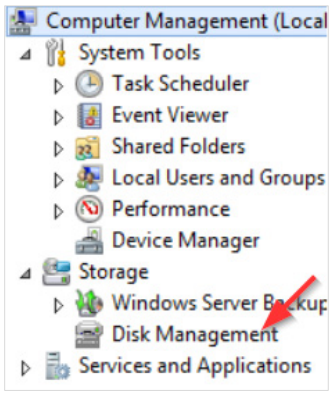


2. From the menu bar, choose **Tools > Computer Management**.

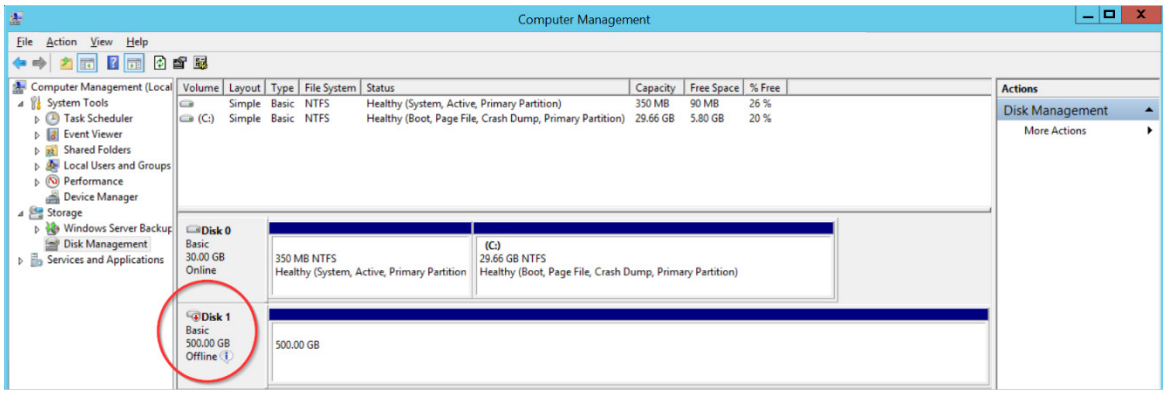
   The Computer Management screen displays.



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



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



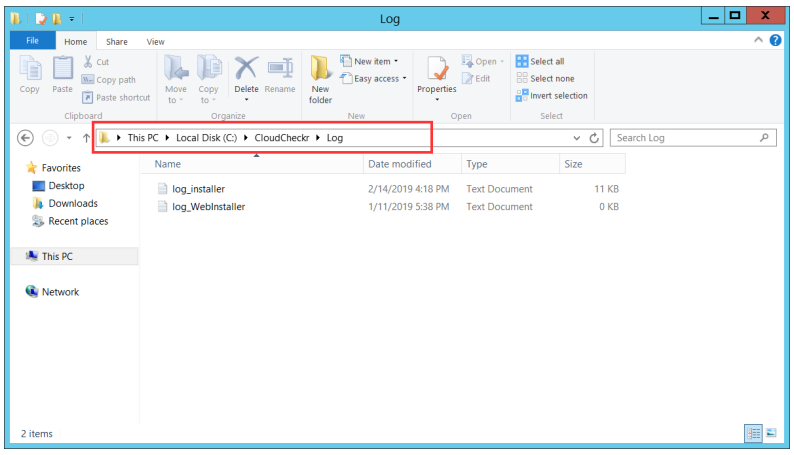
4. Right-click the **disk name** and from the fly-out menu, select **Online**. Your D: drive is now available.

How Do I Access My Log Files?

We 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:

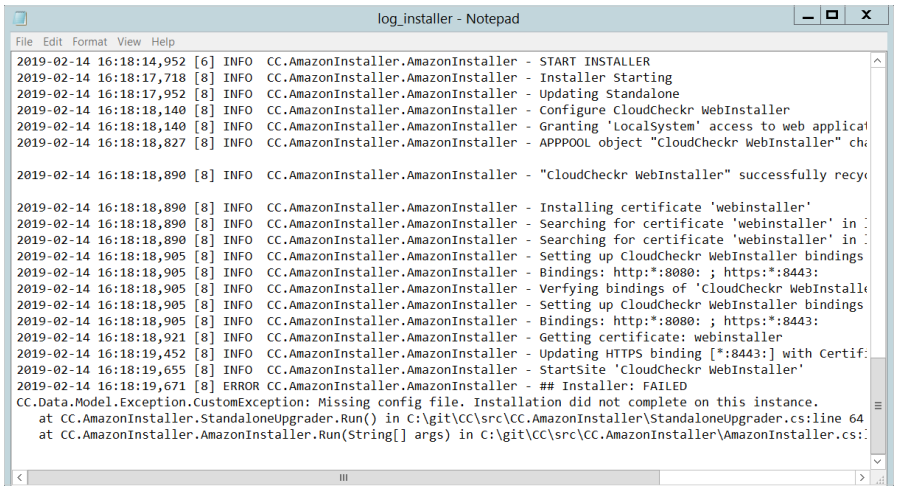
1. From the taskbar, click **Windows Explorer**.

2. 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.



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.

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 #1 (credentials for pricing jobs)	IAM username/access key/secret key
Pricing User #2 (credentials for pricing jobs)	IAM username/access key/secret key
Pricing User #3 (credentials for pricing jobs)	IAM username/access key/secret key
RDS Database	username/password/database name/endpoint/VPC values
S3 Bucket	bucket name/region name/policy/role
EC2 Instance (Web Console)	ID/type/availability zone (region code)
EC2 Instance (Scheduler)	ID/type/availability zone (region code)
EC2 Instance (Workers)	ID/type/availability zone (region code)
Private Key (.PEM) File Location and Name	
Public DNS Name (IPv4)	
Private DNS	
Subnet ID	

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

APPENDIX

IAM Policies

To create a cross-account role manually, copy the least privilege policies, found on the next few pages, and attach them to your role.

Cost

```
{
    "Version":"2012-10-17",
    "Statement":[
       {
          "Sid":"CloudCheckrCostPermissions",
          "Effect":"Allow",
          "Action":[
             "ce:GetReservationUtilization",
             "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":"*"
       }
    ]
 }
```

Billing: DBR

```
{
   "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]/*"
         ]
      }
   ]
}
```

Billing: CUR

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"CostReadCUR",
         "Effect":"Allow",
         "Action":[
                "s3:GetObject"
         ],
         "Resource":[
             "arn:aws:s3:::[YOUR COST AND USAGE REPORT BUCKET]",
             "arn:aws:s3:::[YOUR COST AND USAGE REPORT BUCKET]/*"
         ]
      }
   ]
}
```

Security/Compliance

```
{
    "Version":"2012-10-17",
    "Statement":[
       {
          "Sid":"SecurityPermissons",
          "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 (code block 1 of 3)

Note: Due to length of the Inventory policy, we divided it into three code blocks. Please copy all three code blocks to get the complete Inventory policy.

```
{
    "Version":"2012-10-17",
    "Statement":[
       {
          "Sid":"InventoryAndUtilization",
          "Effect":"Allow",
          "Action":[
             "acm:DescribeCertificate",
             "acm:ListCertificates",
             "acm:GetCertificate",
             "ec2:Describe*",
             "ec2:GetConsoleOutput",
             "autoscaling:Describe*",
             "cloudformation:DescribeStacks",
             "cloudformation:GetStackPolicy",
             "cloudformation:GetTemplate",
             "cloudformation:ListStackResources",
             "cloudfront:List*",
             "cloudfront:GetDistributionConfig",
             "cloudfront:GetStreamingDistributionConfig",
             "cloudhsm:Describe*",
             "cloudhsm:List*",
             "cloudsearch:Describe*",
             "cloudtrail:DescribeTrails",
             "cloudtrail:GetTrailStatus",
             "cloudwatch:DescribeAlarms",
             "cloudwatch:GetMetricStatistics",
             "cloudwatch:ListMetrics",
             "cognito-identity:ListIdentities",
             "cognito-identity:ListIdentityPools",
             "cognito-idp:ListGroups",
             "cognito-idp:ListIdentityProviders",
             "cognito-idp:ListUserPools",
             "cognito-idp:ListUsers",
             "cognito-idp:ListUsersInGroup",
             "config:DescribeConfigRules",
             "config:GetComplianceDetailsByConfigRule",
             "config:Describe*",
```

Inventory (code block 2 of 3)

```
            "datapipeline:ListPipelines",
            "datapipeline:GetPipelineDefinition",
            "datapipeline:DescribePipelines",
            "directconnect:DescribeLocations",
            "directconnect:DescribeConnections",
            "directconnect:DescribeVirtualInterfaces",
            "dynamodb:ListTables",
            "dynamodb:DescribeTable",
            "dynamodb:ListTagsOfResource",
            "ecs:ListClusters",
            "ecs:DescribeClusters",
            "ecs:ListContainerInstances",
            "ecs:DescribeContainerInstances",
            "ecs:ListServices",
            "ecs:DescribeServices",
            "ecs:ListTaskDefinitions",
            "ecs:DescribeTaskDefinition",
            "ecs:ListTasks",
            "ecs:DescribeTasks",
            "ssm:ListResourceDataSync",
            "ssm:ListAssociations",
            "ssm:ListDocumentVersions",
            "ssm:ListDocuments",
            "ssm:ListInstanceAssociations",
            "ssm:ListInventoryEntries",
            "elasticache:Describe*",
            "elasticache:List*",
            "elasticbeanstalk:Describe*",
            "elasticfilesystem:DescribeFileSystem",
            "elasticfilesystem:DescribeTags",
            "elasticloadbalancing:Describe*",
            "elasticmapreduce:Describe*",
            "elasticmapreduce:List*",
            "es:ListDomainNames",
            "es:DescribeElasticsearchDomains",
            "glacier:ListTagsForVault",
            "glacier:DescribeVault",
            "glacier:GetVaultNotifications",
            "glacier:DescribeJob",
            "glacier:GetJobOutput",
            "glacier:ListJobs",
            "glacier:ListVaults",
            "iam:Get*",
            "iam:List*",
            "iam:GenerateCredentialReport",
            "iot:DescribeThing",
            "iot:ListThings",
            "kms:DescribeKey",
            "kms:GetKeyPolicy",
            "kms:GetKeyRotationStatus",
            "kms:ListAliases",
            "kms:ListGrants",
            "kms:ListKeys",
            "kms:ListKeyPolicies",
            "kms:ListResourceTags",
            "kinesis:ListStreams",
            "kinesis:DescribeStream",
            "kinesis:GetShardIterator",
            "kinesis:GetRecords",
```

Inventory (code block 3 of 3)

```
            "lambda:ListFunctions",
            "lambda:ListTags",
            "Organizations:List*",
            "Organizations:Describe*",
            "rds:Describe*",
            "rds:List*",
            "redshift:Describe*",
            "route53:ListHealthChecks",
            "route53:ListHostedZones",
            "route53:ListResourceRecordSets",
            "s3:GetBucketACL",
            "s3:GetBucketLocation",
            "s3:GetBucketLogging",
            "s3:GetBucketPolicy",
            "s3:GetBucketTagging",
            "s3:GetBucketWebsite",
            "s3:GetBucketNotification",
            "s3:GetLifecycleConfiguration",
            "s3:GetNotificationConfiguration",
            "s3:List*",
            "sdb:ListDomains",
            "sdb:DomainMetadata",
            "ses:ListIdentities",
            "ses:GetSendStatistics",
            "ses:GetIdentityDkimAttributes",
            "ses:GetIdentityVerificationAttributes",
            "ses:GetSendQuota",
            "sns:GetSnsTopic",
            "sns:GetTopicAttributes",
            "sns:GetSubscriptionAttributes",
            "sns:ListTopics",
            "sns:ListSubscriptionsByTopic",
            "sqs:ListQueues",
            "sqs:GetQueueAttributes",
            "storagegateway:Describe*",
            "storagegateway:List*",
            "support:*",
            "swf:ListClosedWorkflowExecutions",
            "swf:ListDomains",
            "swf:ListActivityTypes",
            "swf:ListWorkflowTypes",
            "workspaces:DescribeWorkspaceDirectories",
            "workspaces:DescribeWorkspaceBundles",
            "workspaces:DescribeWorkspaces"
        ],
        "Resource":"*"
      }
   ]
}
```

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]/*"
          ]
       }
    ]
 }
```

CloudWatch Flow Logs

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

Deploy CloudCheckr in Additional Availability Zone

Note: This section is optional and recommended for customers who require high availability of the CloudCheckr application.

To ensure that your self-hosted application is available in the event of an outage, we recommend that you install your Web Console EC2 instance and workers in additional availability zone:

1. Follow the steps in the Launch the Self-Hosted AMI section to launch a new EC2 instance.
2. Return to the AWS Management Console.
3. From the AWS Services screen, select **Compute > EC2**.
4. From the dashboard, select **Load Balancing > Load Balancers**.
5. Create a new load balancer, add a new availability zone, and point it to the new EC2 instance by following the AWS instructions, Tutorial: Create a Classic Load Balancer.
6. Install the self-hosted application in the new availability zone following the steps in the Install the Self-Hosted App section.

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