



Self-Hosted Application: Multi-Tier – IC

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 S3 Bucket.....	11
Create an S3 Bucket Policy.....	12
Create an IAM Role.....	14
Launch the Self-Hosted AMI.....	16
Configure the EC2 Instances.....	17
Configure the Web Console.....	17
Configure the Scheduler and Workers.....	22
Install Trusted Certificates.....	23
Provision Your Microsoft SQL Server.....	32
Create an RDS Database (Preferred Method).....	32
Launch a Microsoft SQL Server on an EC2 Instance (Backup Method).....	36
Add Permissions to Your Microsoft SQL Server.....	37
Install the Self-Hosted App.....	40
Install the Web Console.....	40
Input JSON File.....	50
Install the Scheduler.....	52
Install the Workers.....	52
Configure the Self-Hosted App.....	53
License Your App.....	53
Create a Partner.....	55
Complete the Back-End System Configuration.....	58
Create a Trusted User.....	60
Create Trusted User Not in a Standard Region.....	61
Create an Account.....	62
Create Least Privilege Policies.....	64
Create a Cross-Account Role.....	64

Upgrade the Self-Hosted App	68
Frequently Asked Questions	70
Is There an Alternative to Remote Desktop?	70
Why Can't I Open My Browser?	72
Where Is My D: Drive?	73
How Do I Access My Log Files?	75
Required Information	77
Appendix	79
IAM Policies	79
Cost	80
Billing: DBR	81
Billing: CUR	81
Security/Compliance	82
Inventory (code block 1 of 3)	83
Inventory (code block 2 of 3)	84
Inventory (code block 3 of 3)	85
CloudTrail	86
CloudWatch Flow Logs	86
Deploy CloudCheckr in Additional Availability Zone	87

INTRODUCTION

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

The MT-IC version is geared toward customers who cannot operate in the public internet but still require a scalable architecture for larger cloud deployments.

In the MT – IC version, the self-hosted application resides on a cluster of EC2 instances and databases that are completely air-gapped and isolated from the public internet. Communication only occurs between the AWS resources in your deployment and AWS.

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 to 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:

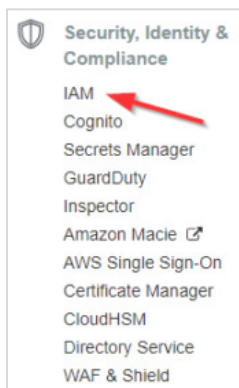
- create an IAM policy that enables the self-hosted application to access AWS pricing data
- create an IAM user and attach them to your pricing policy

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.

The screenshot shows the 'Welcome to Identity and Access Management' dashboard. At the top, it provides an IAM users sign-in link: <https://cloudcheckrdev.signin.aws.amazon.com/console> with a 'Customize' link. Below this, the 'IAM Resources' section displays statistics: Users: 164, Roles: 310, Groups: 81, and Identity Providers: 0. Customer Managed Policies are listed as 318. The 'Security Status' section features a progress bar indicating '4 out of 5 complete'. A list of five security tasks follows, each with a status icon and a dropdown arrow:

Status	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	▼

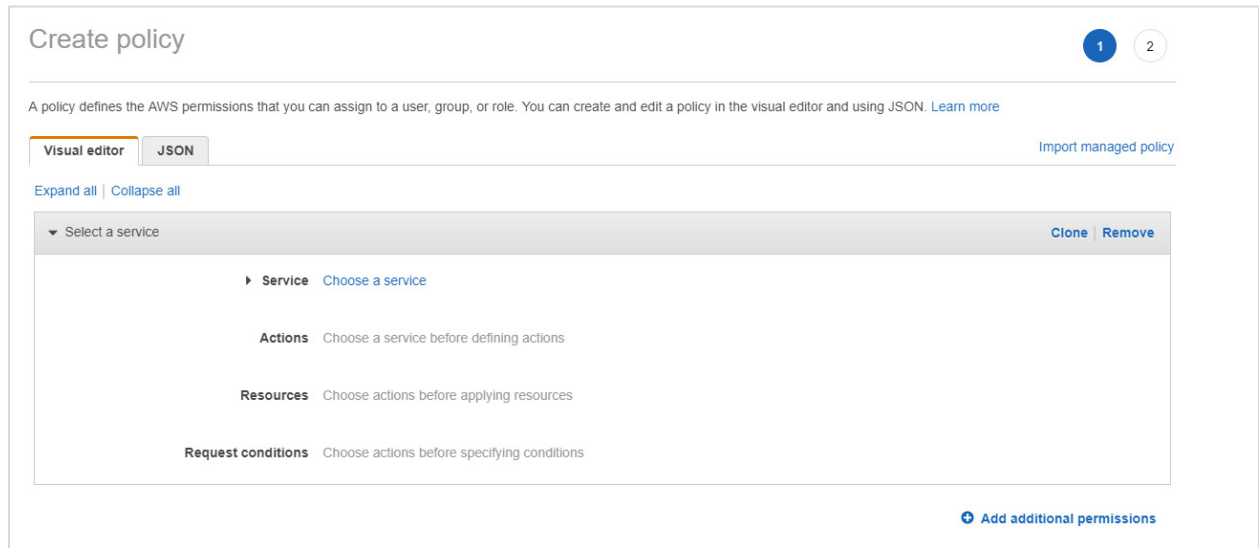
3. From the dashboard, click **Policies**.

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

A list of policies displays.

4. Click **Create policy**.

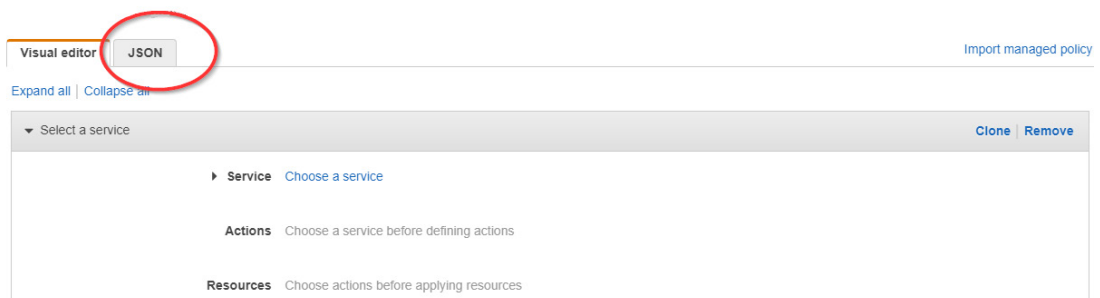
The Create Policy page opens.



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**.

Create policy

Review policy

Name* pricing

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 200 services) Show remaining 199			
EC2	Limited: List	All resources	None

* Required

Cancel Previous **Create policy**

A message indicates that AWS has created your policy.

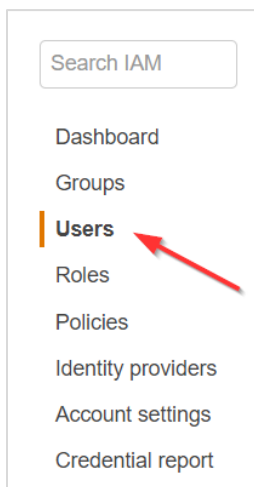
6. Repeat steps 1-5 to create a pricing policy for the remaining two AWS accounts.

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**.

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**

4. Click **Attach existing policies directly**, select your pricing policy, 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**

[Create policy](#)

Filter policies ▼ pricing Showing 1 result

	Policy name	Type	Used as	Description
<input checked="" type="checkbox"/>	pricing	Customer managed	Permissions policy (1)	pricing

Cancel Previous **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**.

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 following policies will be attached to the user shown above.

Type	Name
Managed policy	pricing

Tags

No tags were added.

Cancel Previous **Create user**

A message lets you know that AWS successfully created the user and the 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://cselfhosted.signin.aws.amazon.com/console>

Download .csv

User	Access key ID	Secret access key
pricing_user	AKIAI44QH8DHBEXAMPLE	***** Show

Close

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.

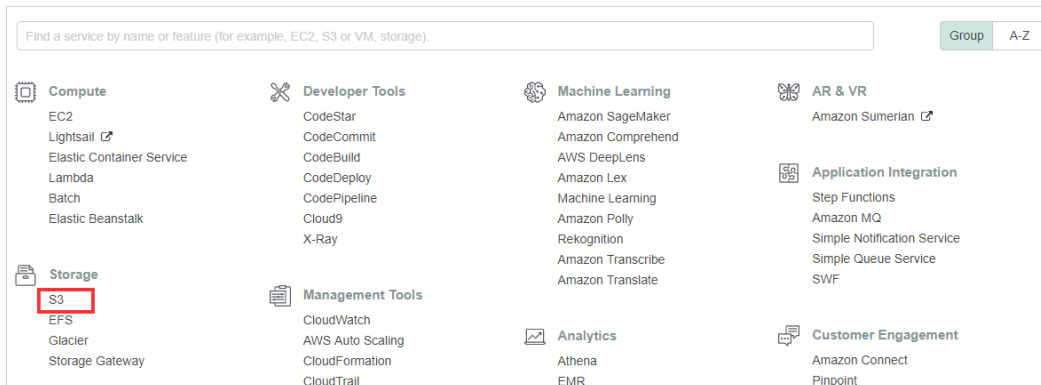
8. Repeat steps 1-7 for the remaining two AWS accounts.

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

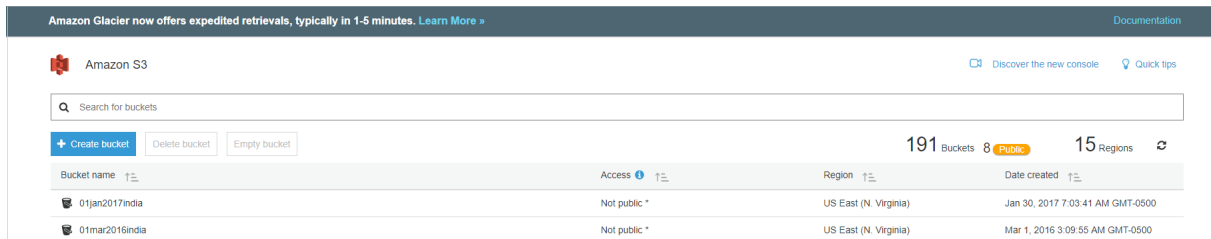
Create an S3 Bucket

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**.



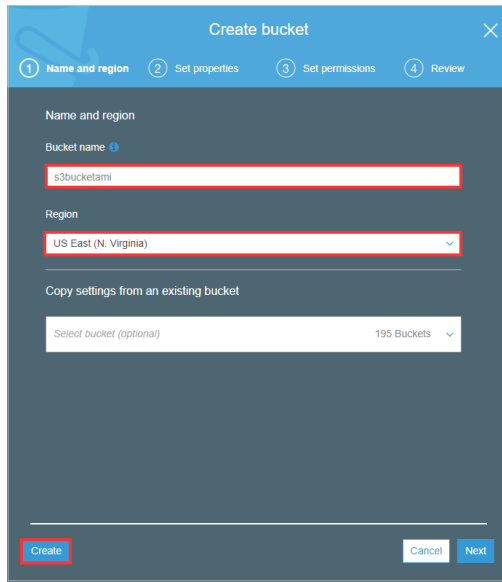
The Amazon S3 page opens.



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 screenshot shows the 'Create bucket' wizard in the AWS console. The title bar says 'Create bucket' with a close button. Below the title bar are four steps: 1. Name and region, 2. Set properties, 3. Set permissions, and 4. Review. The current step is 'Name and region'. It contains three main sections: 'Name and region' with a 'Bucket name' text field containing 's3bucketami', 'Region' with a dropdown menu set to 'US East (N. Virginia)', and 'Copy settings from an existing bucket' with a 'Select bucket (optional)' dropdown menu showing '195 Buckets'. At the bottom, there are three buttons: 'Create' (highlighted in red), 'Cancel', and 'Next'.

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:

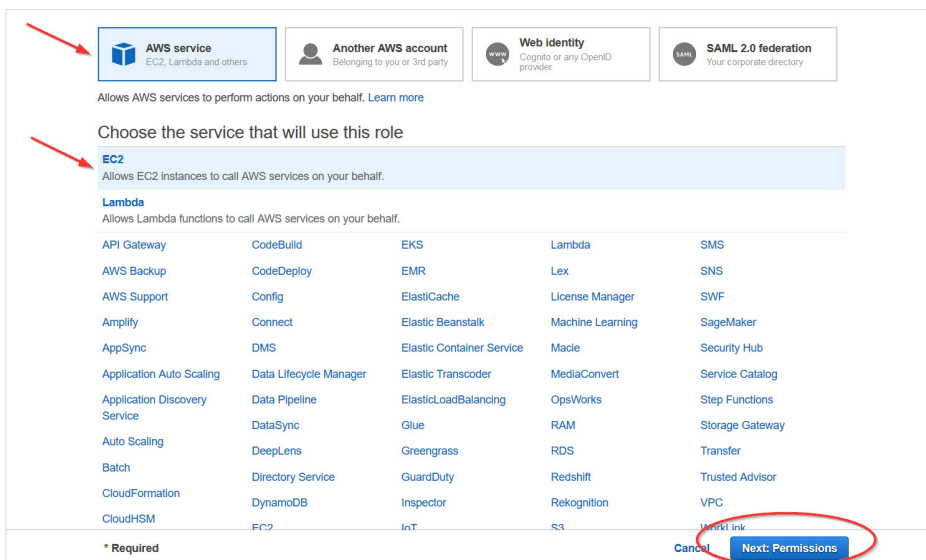
- You can use the wildcards, ***** and **?** within any ARN segment.
- An asterisk, *****, represents a combination of zero or more characters.
- A question mark, **?**, represents a single character.
- You can use multiple ***** and **?** in each segment, but a wildcard cannot span segments.

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**.

Create role

1 2 3 4

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy

Filter policies Q s3 Showing 6 results

	Policy name	Used as	Description
<input type="checkbox"/>	AmazonDMSRedshiftS3Role	None	Provides access to manage S3 settings f...
<input type="checkbox"/>	AmazonS3FullAccess	None	Provides full access to all buckets via th...
<input type="checkbox"/>	AmazonS3ReadOnlyAccess	None	Provides read only access to all buckets...
<input type="checkbox"/>	QuickSightAccessForS3StorageManagement...	None	Policy used by QuickSight team to acces...
<input type="checkbox"/>	S3bucketpolicy	None	
<input type="checkbox"/>	S3bucketpolicy2	None	

▶ Set permissions boundary

* Required Cancel Previous **Next: Tags**

5. Click **Next: Review**.

The Create role 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 AWS service: ec2.amazonaws.com

Policies S3bucketpolicy

Permissions boundary Permissions boundary is not set

No tags were added.

* Required Cancel Previous **Create role**

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

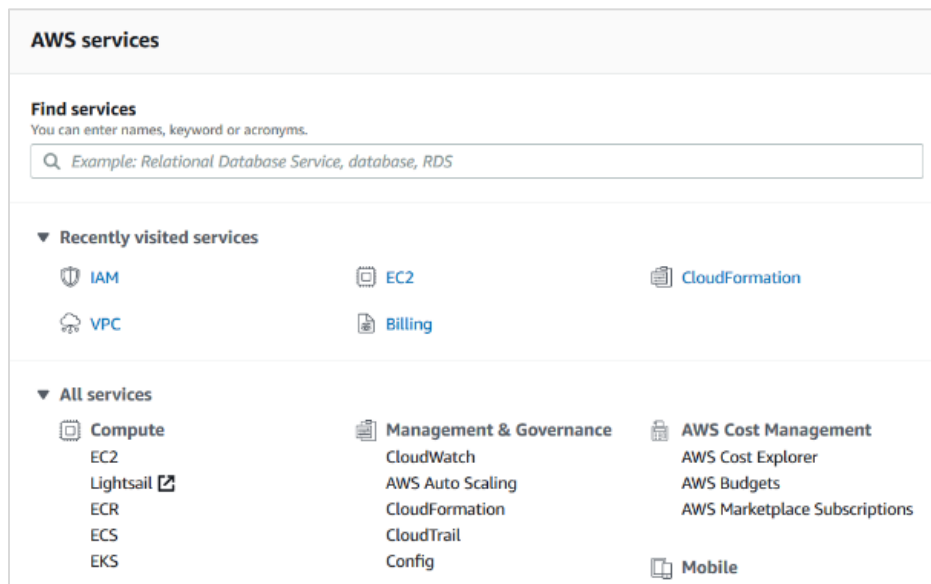
7. Copy the name of the role that will have access to the S3 bucket to the Required Information section.

LAUNCH THE SELF-HOSTED AMI

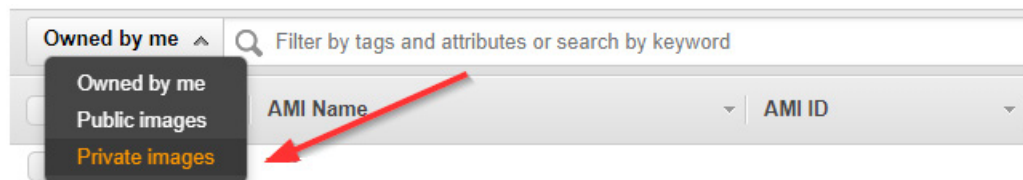
CloudCheckr will let you know when the self-hosted **Amazon Machine Image (AMI)** is available in the AWS Management Console. The AMI contains all the information you need to launch your **EC2 instance**, which is the virtual server where you will run the self-hosted application.

1. Log in to the AWS Management Console.

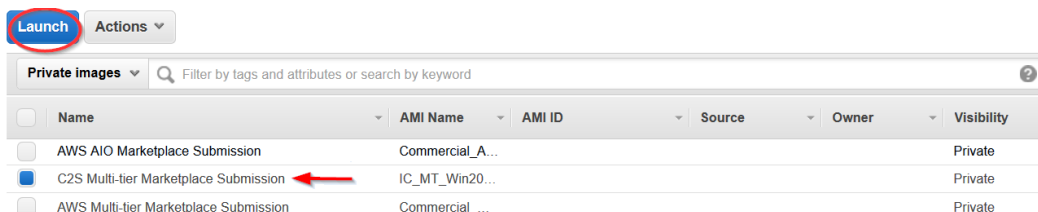
The AWS services page opens.



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.

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 (GiB)	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 Free tier eligible	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	t2.2xlarge	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 Gigabit	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

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. Verify that you have the right volume (EBS) and device (xvdf). This volume acts as the D: drive for your self-hosted application.

- Click **Next: Add Tags**.

Step 5: Add Tags displays. For the purposes of this procedure, we will not add any tags.

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

Step 6: Configure Security Group displays. Configure the security group that will control traffic in and out of your EC2 instance.

- Select one of the configuration methods:

To create a new security group:

- Select the **Create a new security group** radio button.
- 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
6	Required to access your Microsoft SQL server	MS SQL	TCP	1433	Your IP address

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
MS SQL	TCP	1433	My IP
RDP	TCP	3389	My IP
HTTP	TCP	80	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
HTTPS	TCP	443	Custom 0.0.0.0, ::/0

To use an existing security group:

- Select the **Select an existing security group** radio button.
- 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
<input type="checkbox"/> sg-3474bb7c	default	default VPC security group
<input checked="" type="checkbox"/> sg-078381c		ELB created security group
<input type="checkbox"/> sg-034b5fd5		launch-wizard-1

- 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](#)

Commercial
 AWS Multi-tier
 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
m5.xlarge	15	4	16	EBS only	Yes	Up to 10 Gigabit

Security Groups [Edit security groups](#)

Security group name launch-wizard-12
Description launch-wizard-12 created 2019-02-20T13:03:28.708-05:00

[Cancel](#) [Previous](#) [Launch](#)

- 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 X

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

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.

10. Click **Launch Instances**.

11. 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.

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

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

Description	
Instance ID	[REDACTED]
Instance state	running
Instance type	t3.small
Elastic IPs	3.90.130.240*
Availability zone	us-east-1a
Security groups	Web, Jump, view inbound rules, view outbound rules
Scheduled events	No scheduled events
AMI ID	Cannot load details for ami-02554f8b14ce4f6a7. You may not be permitted to view it.
Platform	windows
IAM role	selfhosted-aio-12-4-0-4-Ec2Role
Key pair name	selfhosted
Owner	[REDACTED]
Launch time	January 29, 2019 at 11:39:09 AM UTC-5 (171 hours)
Termination protection	False
Lifecycle	normal
Public DNS (IPv4)	[REDACTED] compute-1.amazonaws.com
IPv4 Public IP	3.90.130.240
IPv6 IPs	-
Private DNS	[REDACTED]
Private IPs	[REDACTED]
Secondary private IPs	
VPC ID	[REDACTED]
Subnet ID	subnet-[REDACTED]
Network interfaces	eth0
Source/dest. check	True
T2/T3 Unlimited	Enabled
EBS-optimized	True
Root device type	ebs
Root device	/dev/sda1
Block devices	/dev/sda1

14. 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

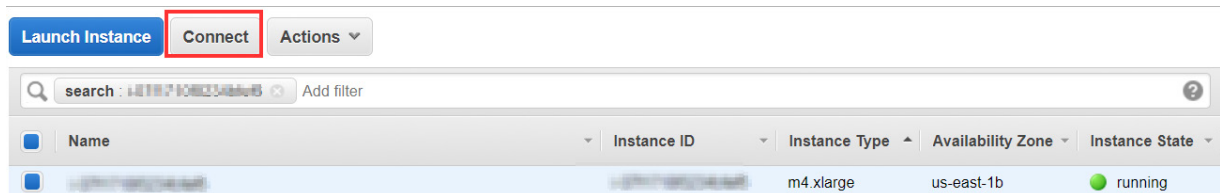
To configure the Scheduler EC2 instance and the Workers EC2 instance, follow the instructions in the [Configure the Web Console](#) section **except** use **m5.large** for your instance type.

INSTALL TRUSTED CERTIFICATES

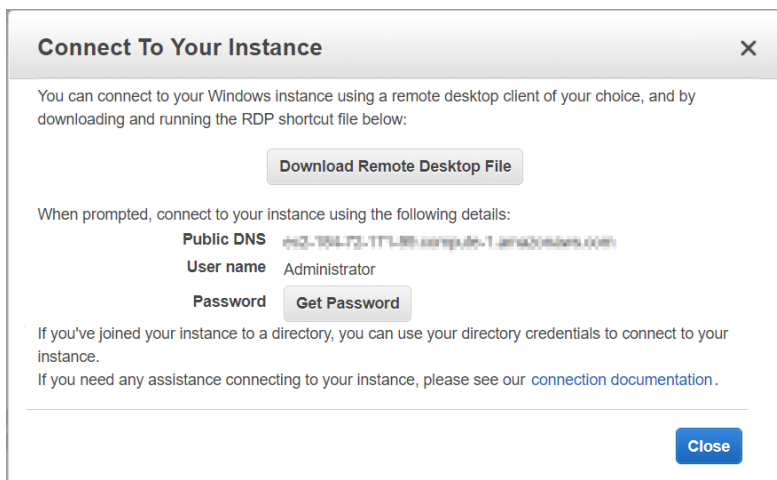
To authenticate to the IC region, you must install trusted certificates on the CloudCheckr EC2 instance. Because these certificates are classified, CloudCheckr cannot install them on the AMI prior to delivery. Your organization can provide you with the required **root certificate** and an **intermediate certificate**, so you can install them onto the server remotely.

Note: You can access the EC2 instance remotely many ways. This procedure shows one way to access the EC2 instance remotely.

1. Before you install the certificates, verify that you have the following prerequisites:
 - Remote Desktop Protocol (RDP)
 - EC2 instance ID
 - Public DNS name of the instance
 - .PEM file
2. From the Amazon Management Console, find the selected instance and click **Connect**.

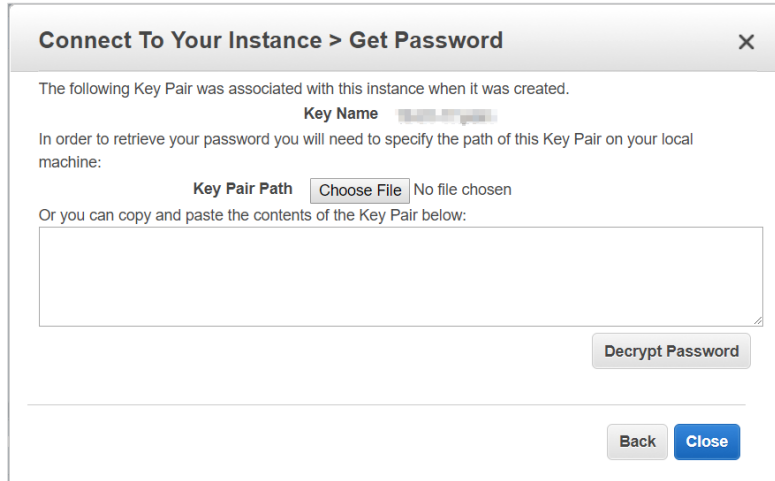


The Connect To Your Instance dialog box opens.

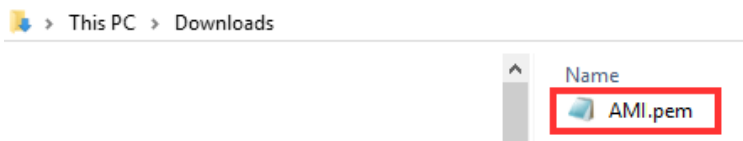


3. Click **Get Password**.

The Connect Your Instance > Get Password dialog box opens.

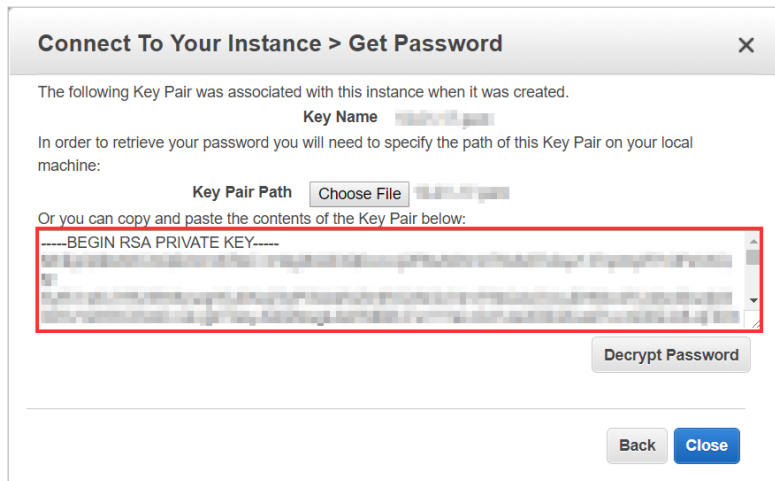


4. Click **Choose File** and navigate to the desktop location where you saved the .PEM file.



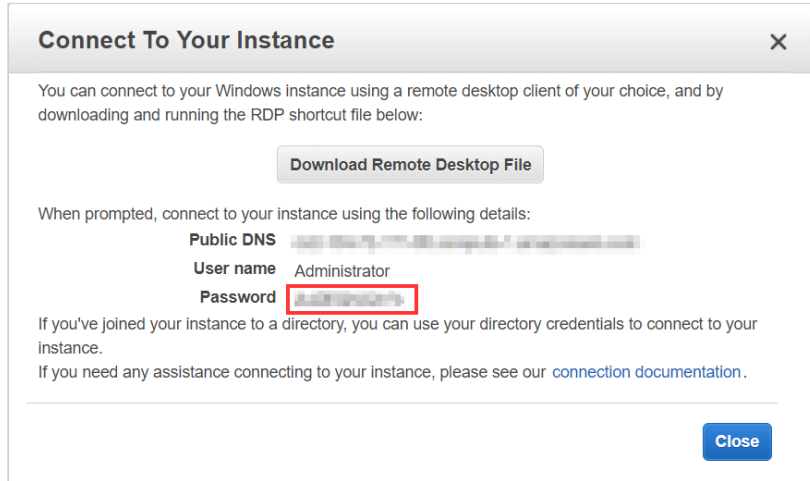
5. Click **Open**.

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



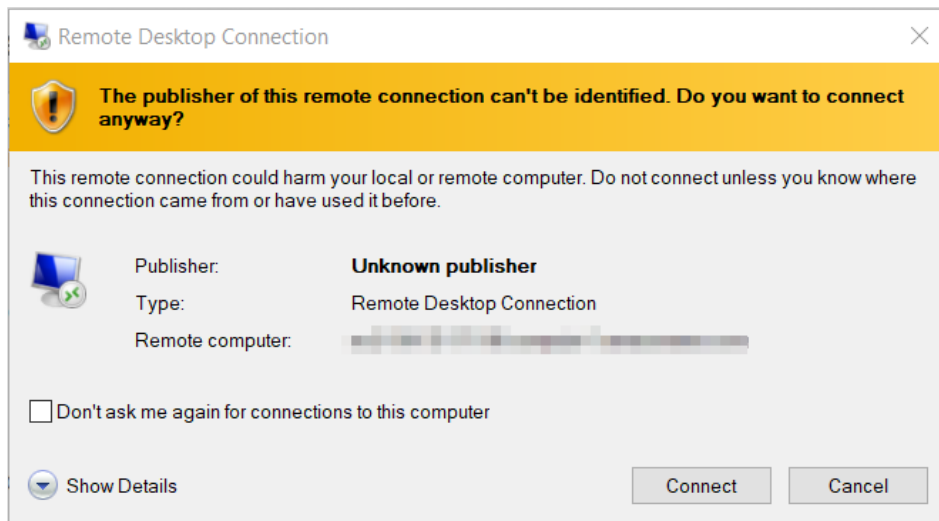
6. Click **Decrypt Password**.

The default administrator password displays.



7. Copy and store this password in a safe place.
8. Click **Download Remote Desktop File**.
9. Open or save the .RDP file.

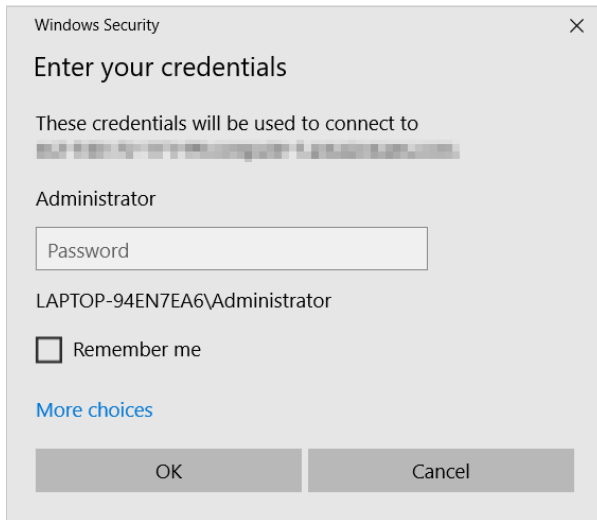
The Remote Desktop Connection dialog box opens.



10. Click **Connect**.

Note: If you receive a warning that the security certificate could not be authenticated, you can verify the identity of the remote computer or just click **Connect**.

The Windows Security dialog box opens.

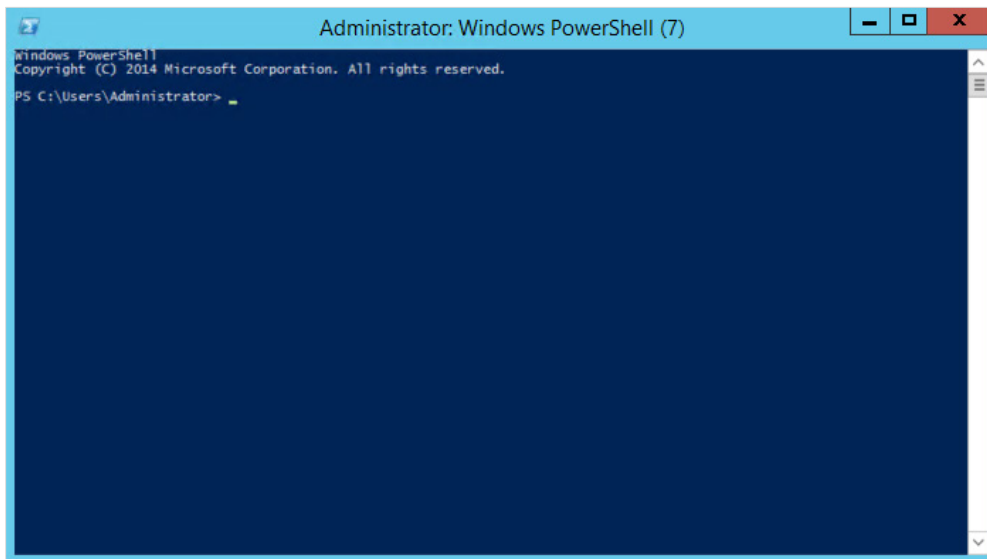


11. Paste the password you copied earlier into the Administrator text field and click **OK**.

Your remote desktop session opens.

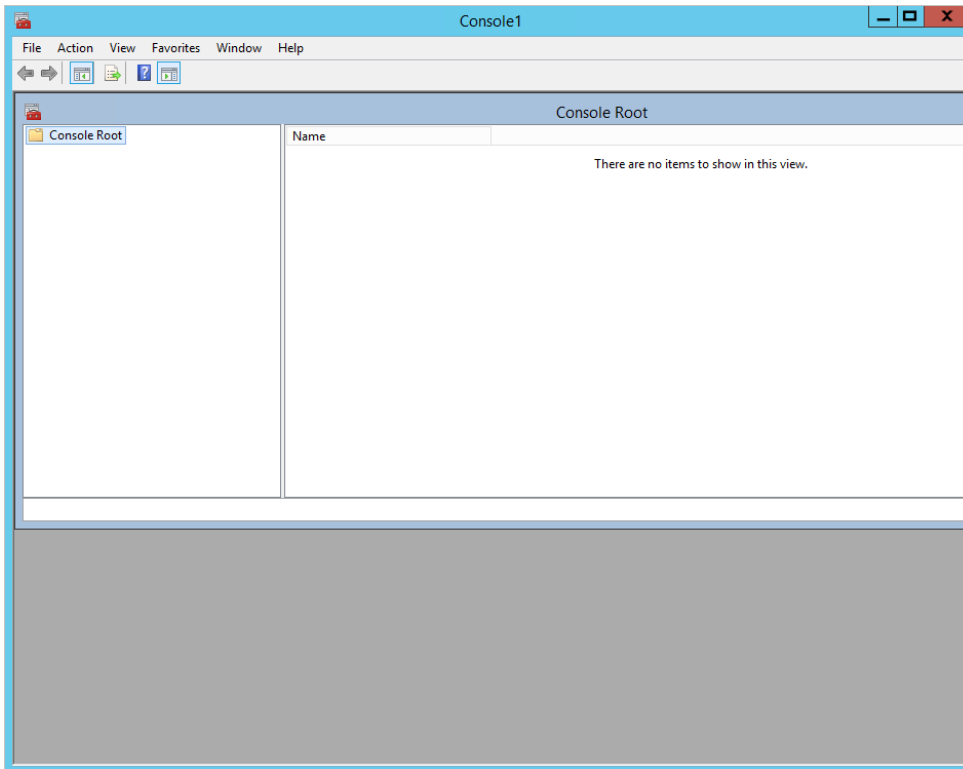
12. In the taskbar, click  (Windows PowerShell icon).

The Administrator: Windows PowerShell command window opens.



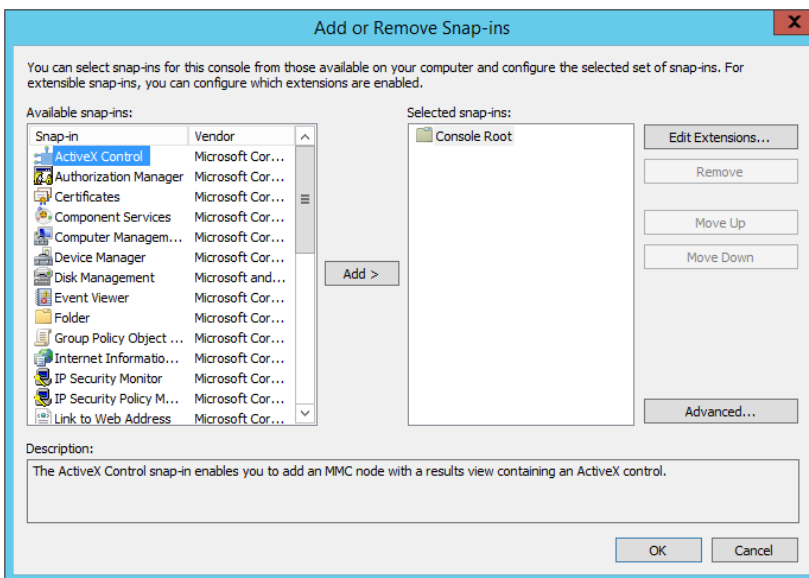
13. At the command prompt, type **MMC**

The MMC console opens.

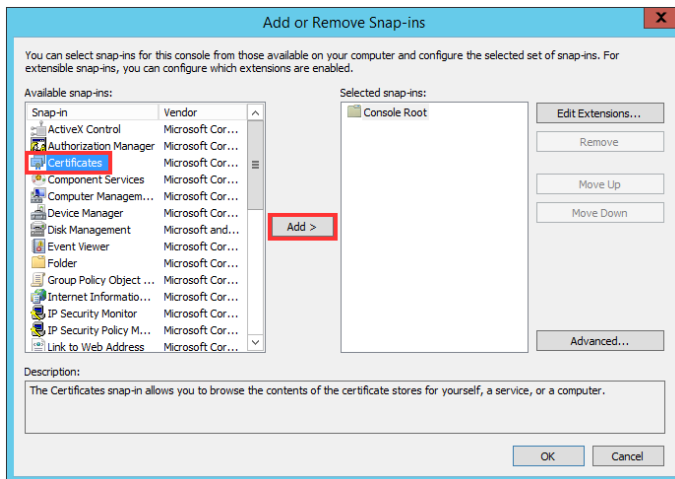


14. From the menu bar, choose **File > Add/Remove Snapin**.

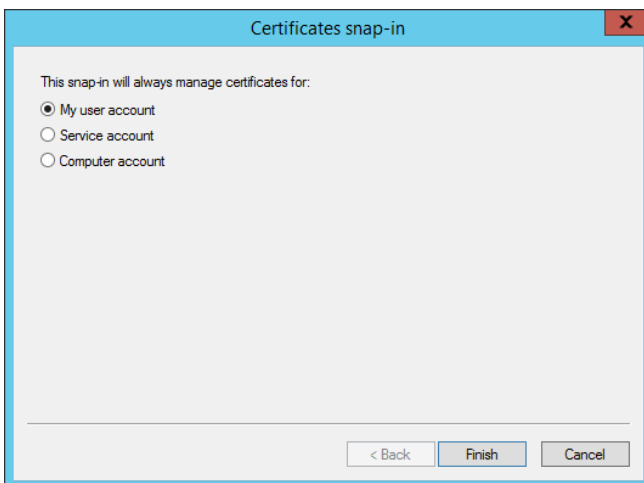
The Add or Remove Snap-ins dialog box opens.



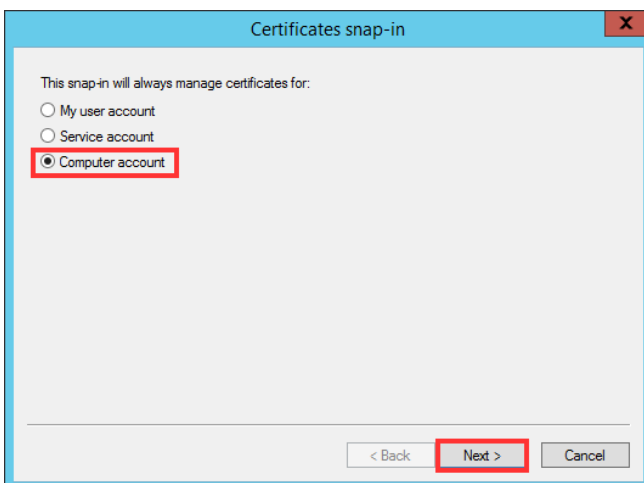
15. From the Available snap-ins list, select **Certificates** and click **Add**.



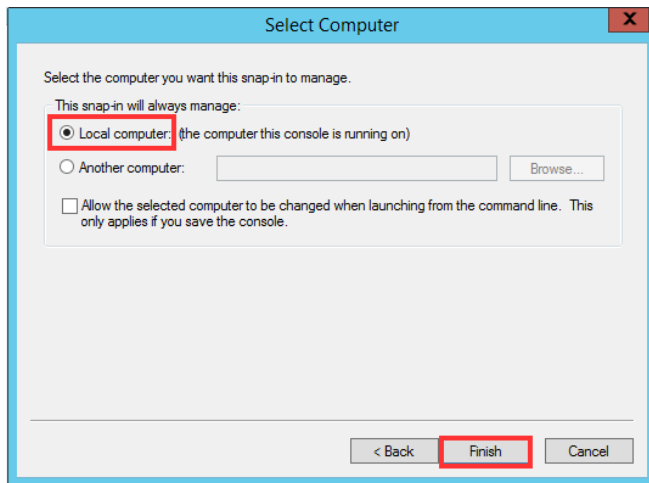
The Certificates snap-in dialog box opens.



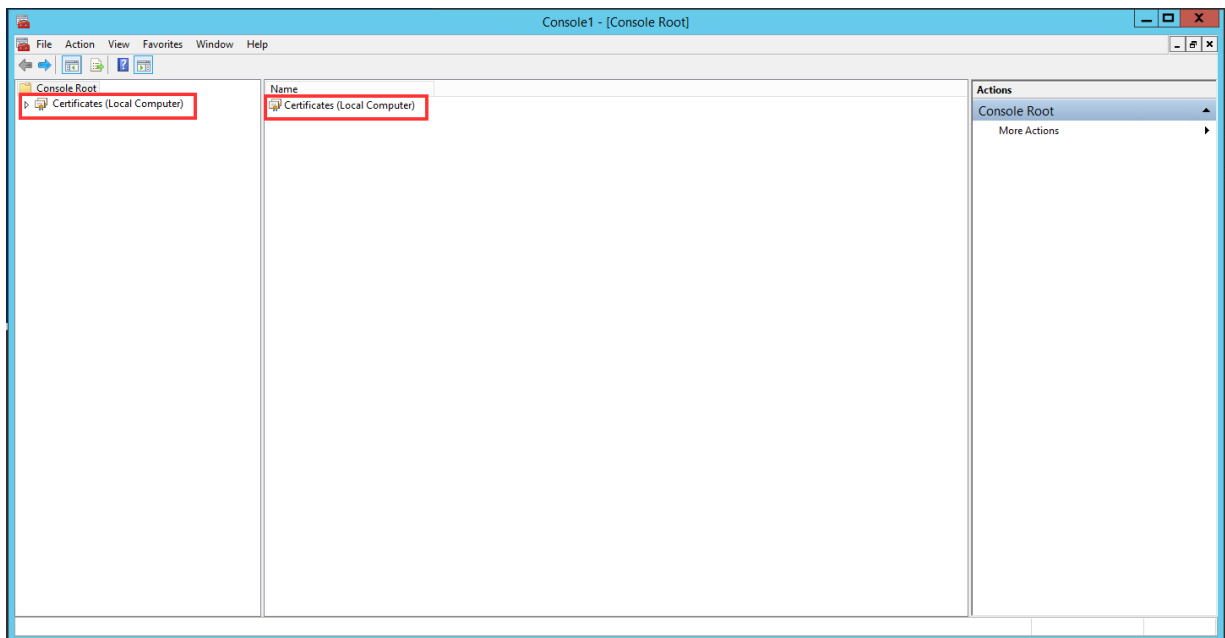
16. Select **Computer account** and click **Next**.



17. Select **Local computer** and click **Finish**.

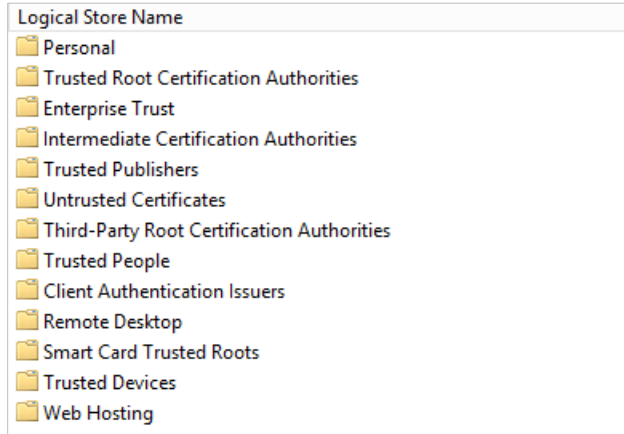


The Add or Remove Snap-ins dialog box opens displaying **Certificates (Local Computer)** in the left and middle panels.

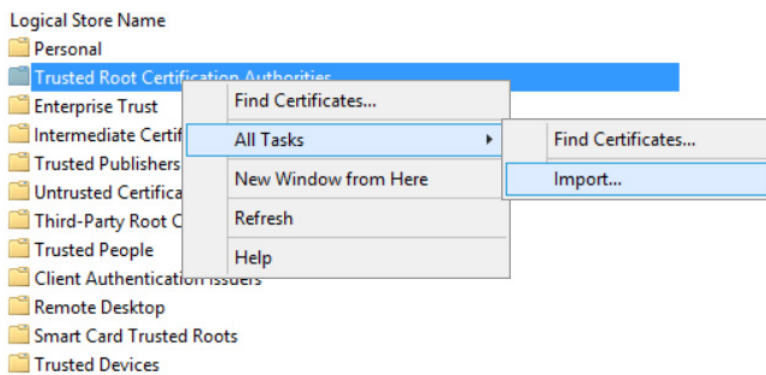


18. From the left panel, click **Certificates (Local Computer)**.

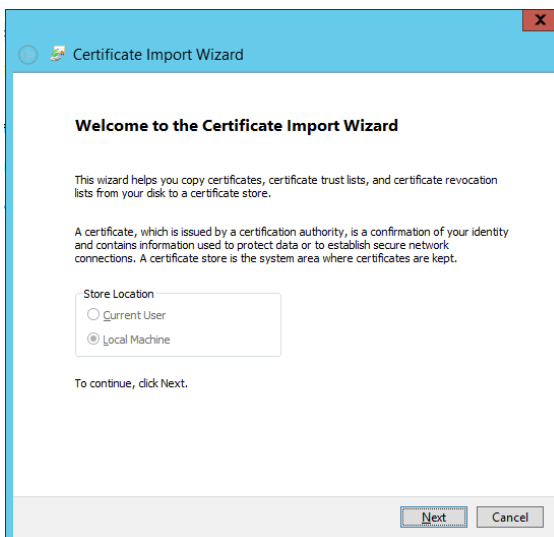
The middle panel displays a list of options.



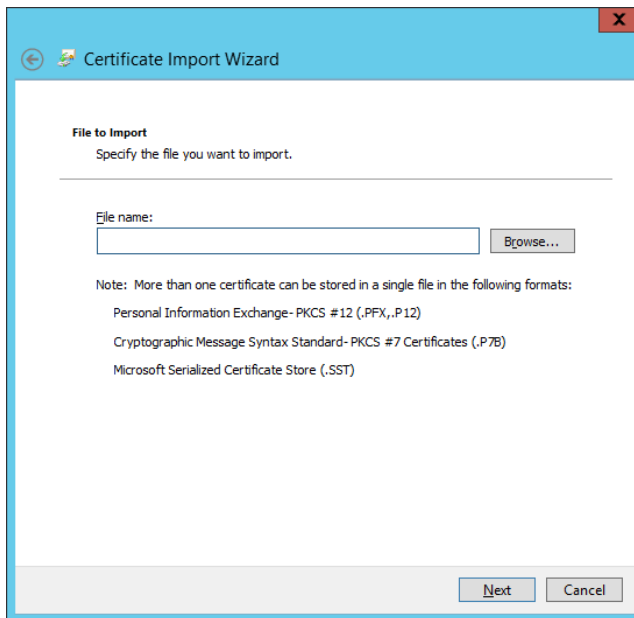
19. From the middle panel, right-click **Trusted Root Certification Authorities** and from the fly-out menus, select **All Tasks > Import**.



The Certificate Import Wizard opens. A **certificate store** is the system location where certificates are kept. The wizard indicates that the default certificate store location is your local machine.



20. Click **Next** to accept the default store location and continue with the wizard.
21. Type the full path and file name of the file you want to import or click **Browse** to navigate to the file.
22. Click **Next**.



23. Click **Finish** to complete the wizard.
A message indicates that your root certificate imported successfully.
24. From the middle panel, right-click **Intermediate Certification Authorities** and from the fly-out menus, select **All Tasks > Import**.
25. Follow the steps in the wizard to import the intermediate certificate successfully.

PROVISION YOUR MICROSOFT SQL SERVER

Once you have created your AWS credentials and configured your EC2 instances, you will need to provision a Microsoft SQL server where AWS can store your data.

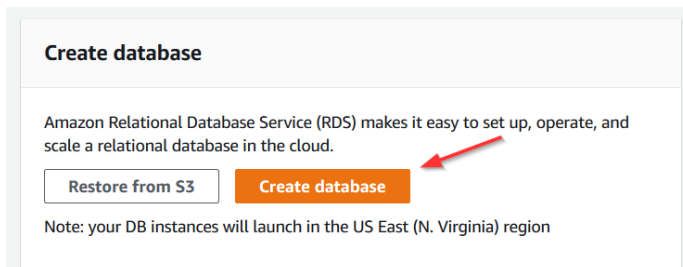
Create an RDS Database (Preferred Method)

The preferred method for provisioning a Microsoft SQL server is to create a **Relational Database Service (RDS)** database. You must set up the RDS database before installation because the Web installer will prompt you to connect to the server during installation.

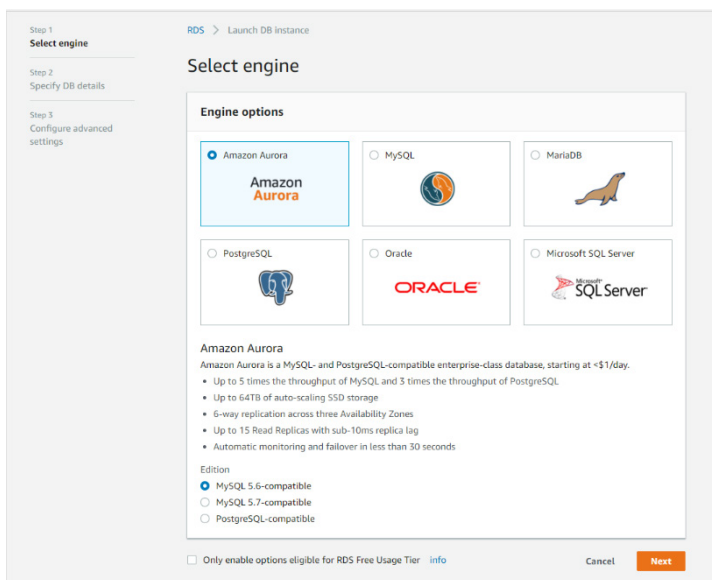
1. Return to the AWS Services page.
2. From the Database section, select **RDS**.

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**.

PostgreSQL

Oracle

Microsoft SQL Server

Microsoft SQL Server

Edition

SQL Server Express Edition
Affordable database management system that supports database sizes up to 10 GB.

SQL Server Web Edition
In accordance with Microsoft's licensing policies, it can only be used to support public and Internet-accessible webpages, websites, web applications, and web services.

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.

DB engine
Microsoft SQL Server Web Edition

License model [info](#)
license-included

DB engine version [info](#)
SQL Server 2012 11.00.6020.0.v1

Free tier
The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions [here](#).
The database engine or edition you selected is not eligible for RDS Free Tier.

DB instance class [info](#)
db.m4.large — 2 vCPU, 8 GiB RAM

Time zone (optional)
No preference

Storage type [info](#)
General Purpose (SSD)

Allocated storage
500 GB
(Minimum: 20 GB, Maximum: 16384 GB) Higher allocated storage [may improve](#) IOPS performance.

9. Scroll down to the Settings section.

Settings

DB instance identifier [info](#)
Specify a name that is unique for all DB instances owned by your AWS account in the current region.

DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance".
Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server).
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Master username [info](#)
Specify an alphanumeric string that defines the login ID for the master user.

Master Username must start with a letter. Must contain 1 to 64 alphanumeric characters.

Master password [info](#) **Confirm password** [info](#)

Master Password must be at least eight characters long, as in "mypassword". Can be any printable ASCII character except "/", "", or "@".

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.

Configure advanced settings

Network & Security

Virtual Private Cloud (VPC) [info](#)
VPC defines the virtual networking environment for this DB instance.

Only VPCs with a corresponding DB subnet group are listed.

Subnet group [info](#)
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

Public accessibility [info](#)

Yes
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

No
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

Availability zone [info](#)

VPC security groups
Security groups have rules authorizing connections from all the EC2 instances and devices that need to access the DB instance.

Create new VPC security group

Choose existing VPC security groups

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.

The screenshot shows the AWS RDS console for a database instance named 'selfhosted-mt'. The 'Connectivity & security' tab is active, displaying the following details:

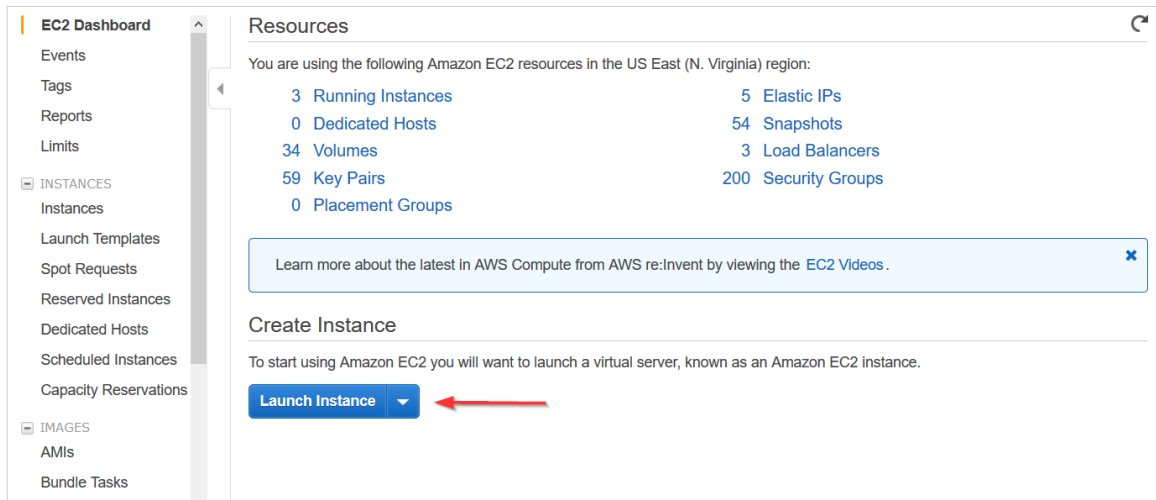
Endpoint & port	Networking	Security
Endpoint selfhosted- mt [redacted] us-east-1.rds.amazonaws.com	Availability zone us-east-1a	VPC security groups Database (sg-[redacted]) (active) Jump (sg-[redacted]) (active)
Port 1433	VPC selfhosted- mt VpcStack-[redacted]-Vpc (vpc-[redacted])	Public accessibility Yes
	Subnet group	

17. Copy the endpoint and VPC values to the [Required Information](#) section.

Launch a Microsoft SQL Server on an EC2 Instance (Backup Method)

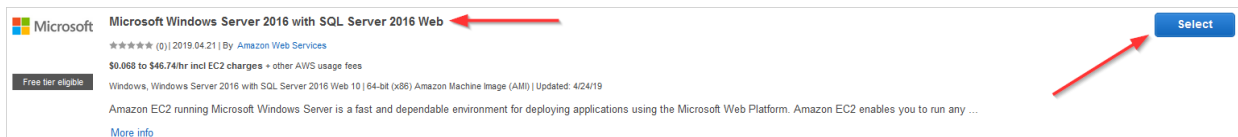
If your company does not have access to Microsoft RDS, you can launch an EC2 instance that will house your Microsoft SQL server. **Use this method as backup method only.**

1. Return to the EC2 dashboard in the AWS Management Console.
2. From the middle of the page, click **Launch Instance**.



Step 1: Choose an Amazon Machine Image (AMI) of the configuration wizard opens.

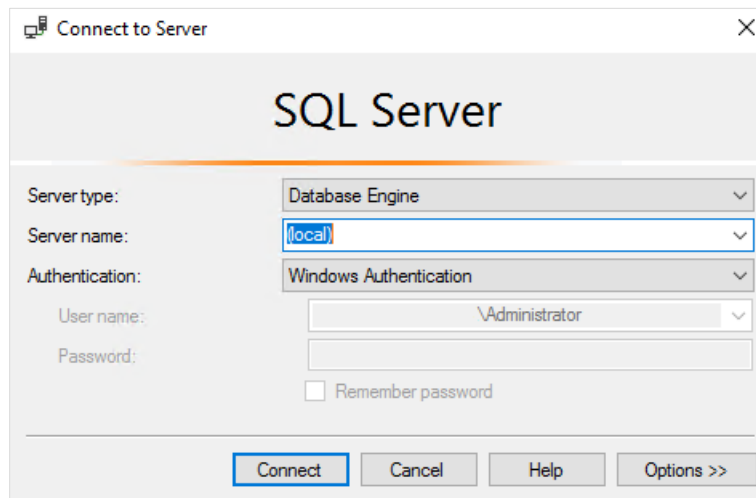
3. In the Search bar, type **SQL Server 2016** to display the associated AMIs.
4. Locate the **Microsoft Windows Server 2016 with SQL Server 2016 Web** and click **Select**.



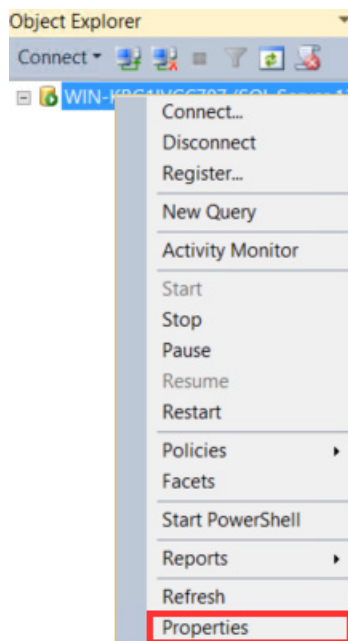
5. Repeat the configuration steps from the [Configure the Web Console](#) section to launch your new EC2 instance with Microsoft SQL Server 2016.
6. After the instance launches, copy the private IP of the SQL server to the [Required Information](#) section.

Add Permissions to Your Microsoft SQL Server

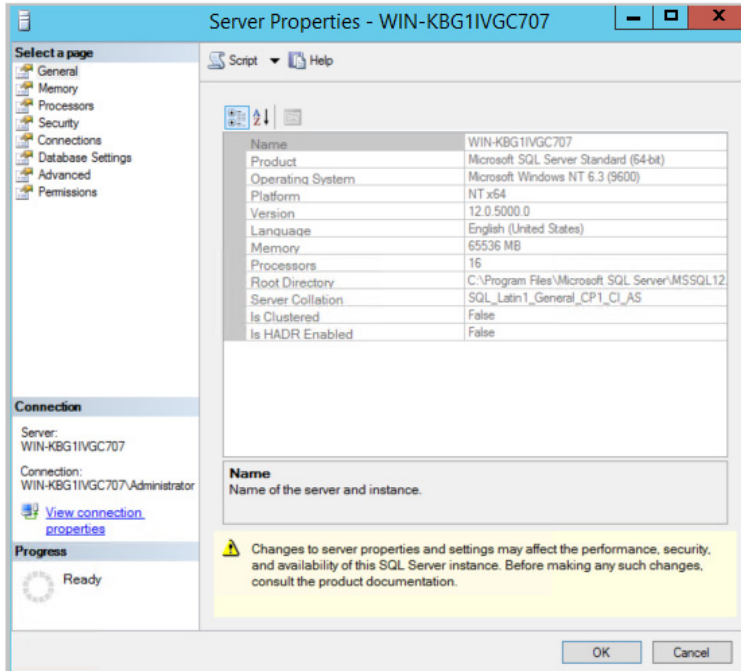
1. Remote desktop into your SQL server instance by following steps 1-12 in the [Install Trusted Certificates](#) section.
2. Type **ssms.exe** at the prompt to open SQL Server Management Studio.
3. Select the server type, server name, and click **Connect**.



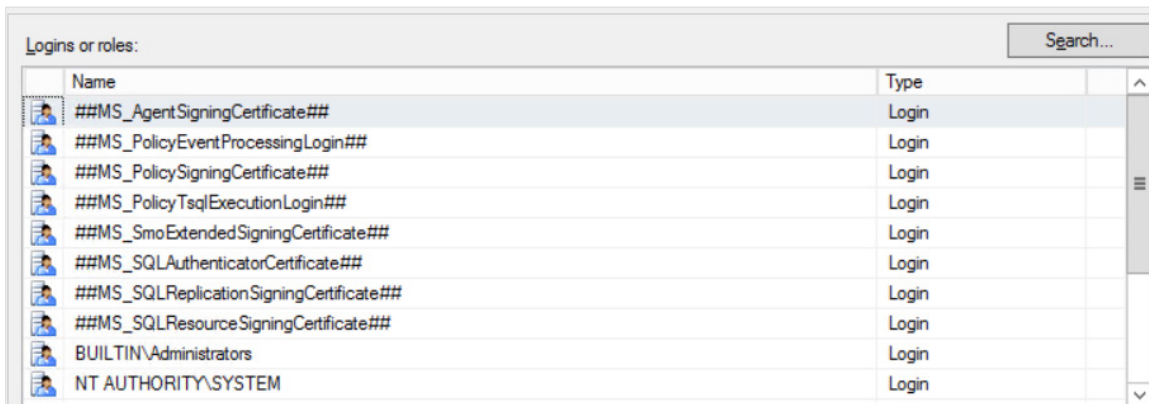
4. From the Object Explorer task pane, right-click the **server name** and select **Properties**.



The Server Properties dialog box opens.



5. Click **Permissions**.
6. Under the Logins or roles section, select the role you want to associate with your SQL server.



- Under the Permissions section, select the user permissions you want to associate with the role.

Permissions for ##MS_SQLResourceSigningCertificate##:

Permission	Grantor	Grant	With Grant	Deny
Administer bulk operations		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alter any availability group		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alter any connection		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alter any credential		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alter any database		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alter any endpoint		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alter any event notification		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alter any event session		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alter any linked server		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- For each permission that you select, select the check box for the type of permission (**Grant**, **With Grant**, or **Deny**).

Note: The role can have Grant and With Grant permissions simultaneously. If you select **Deny**, this is the only permission type that will be available to the role.

Permission	Grantor	Grant	With Grant	Deny
Administer bulk operations		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alter any availability group		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Alter any connection		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Click **Security**.
- Under Server authentication, select the **SQL Server and Windows Authentication mode**.

Server authentication

Windows Authentication mode

SQL Server and Windows Authentication mode

- Click **OK**.

INSTALL THE SELF-HOSTED APP

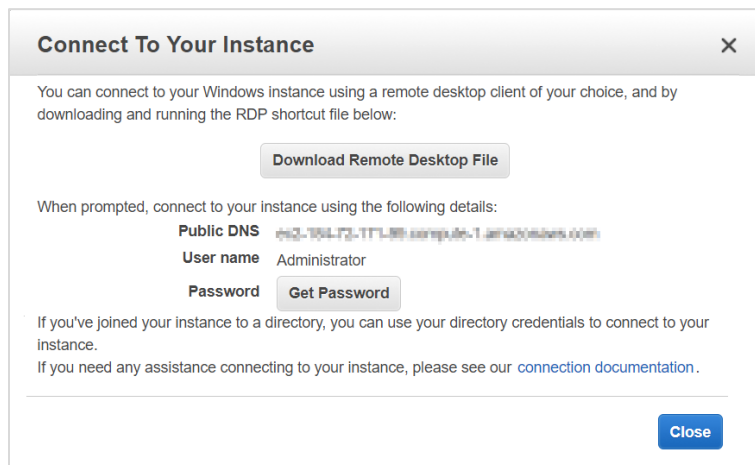
This section shows you how to install a separate EC2 instance for the Web Console, Scheduler, and Workers.

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.

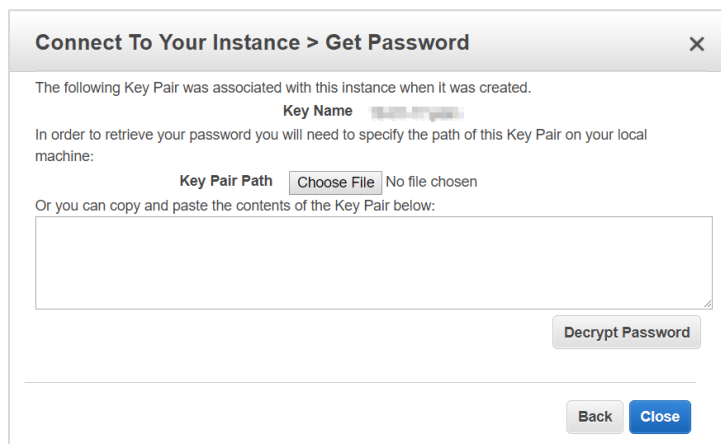
By connecting to each EC2 instance through a Remote Desktop session, you can better manage the installation process and troubleshoot any issues that may occur.

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.

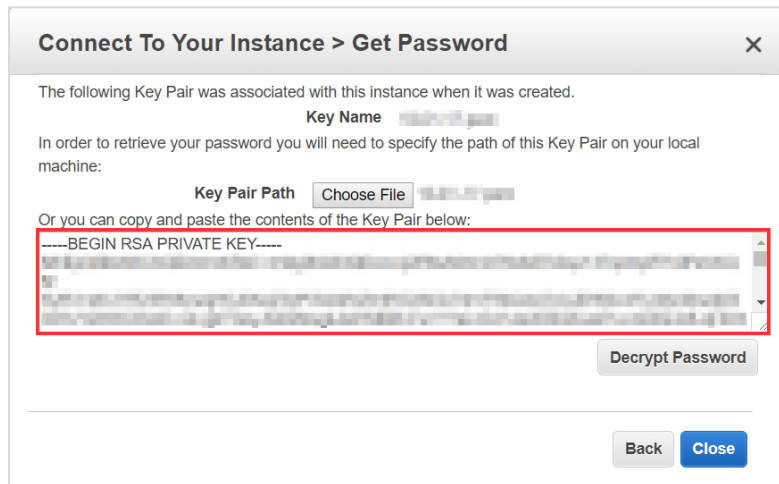


3. Click **Get Password**. The Connect Your Instance > Get Password dialog box opens.



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.

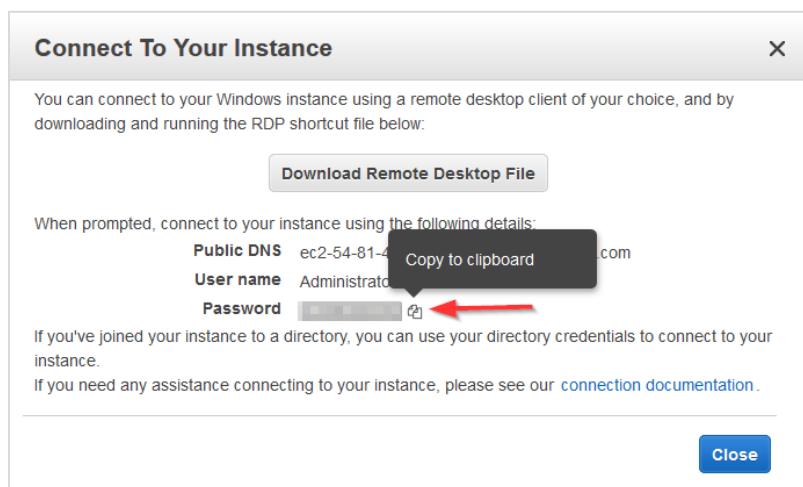


6. Click **Decrypt Password**.

The default administrator password displays.

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

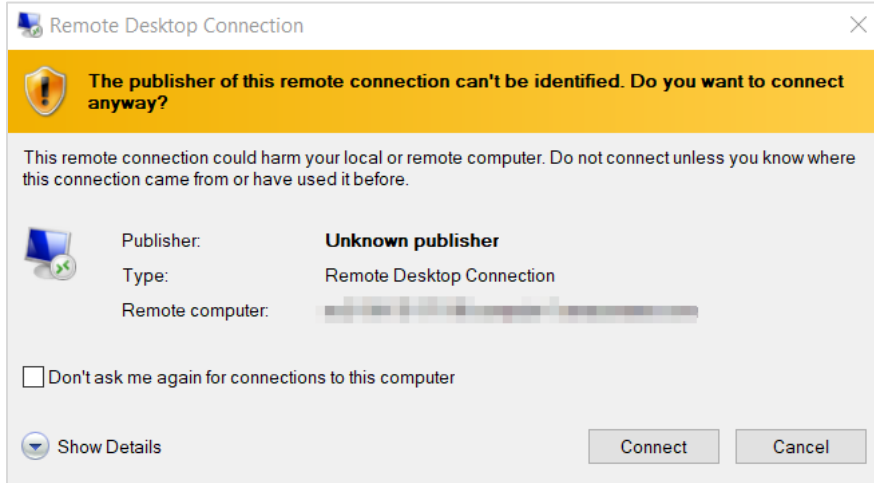
8. Click  to save the password.



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

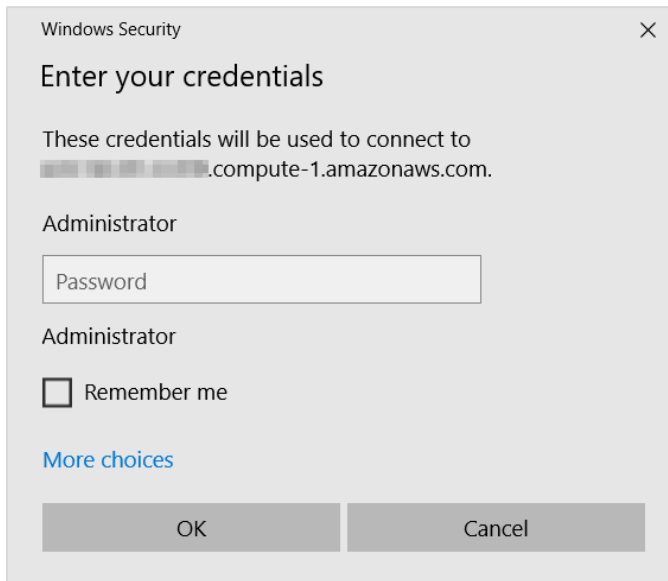
10. Open or save the .RDP file.

The Remote Desktop Connection dialog box opens.



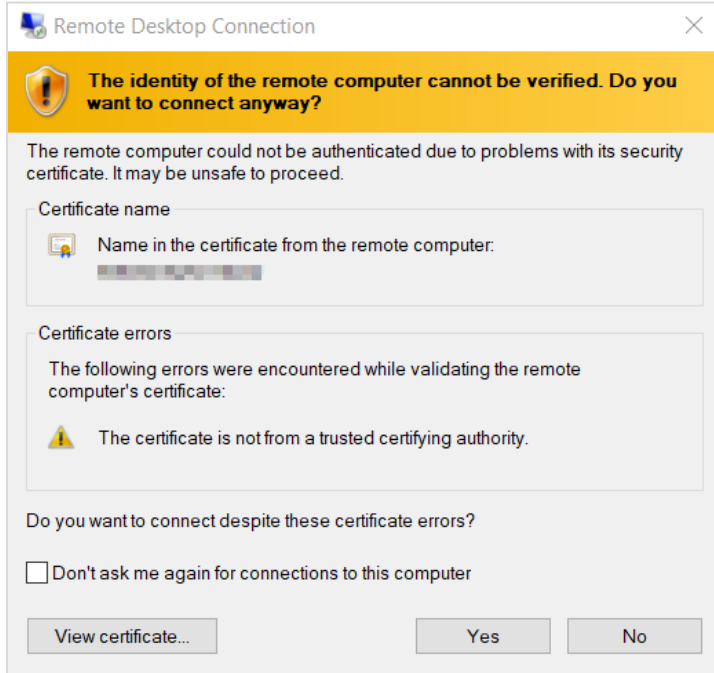
11. Click **Connect**.

The next dialog box prompts you to provide your password.



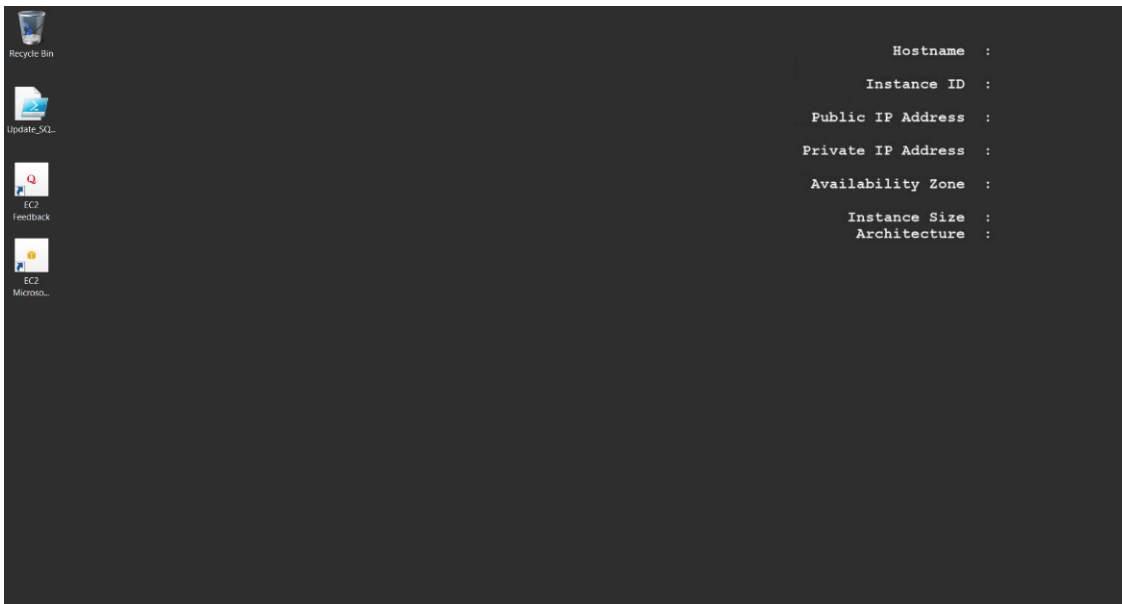
12. In the Administrator text field, paste the password you copied earlier and click **OK**.

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

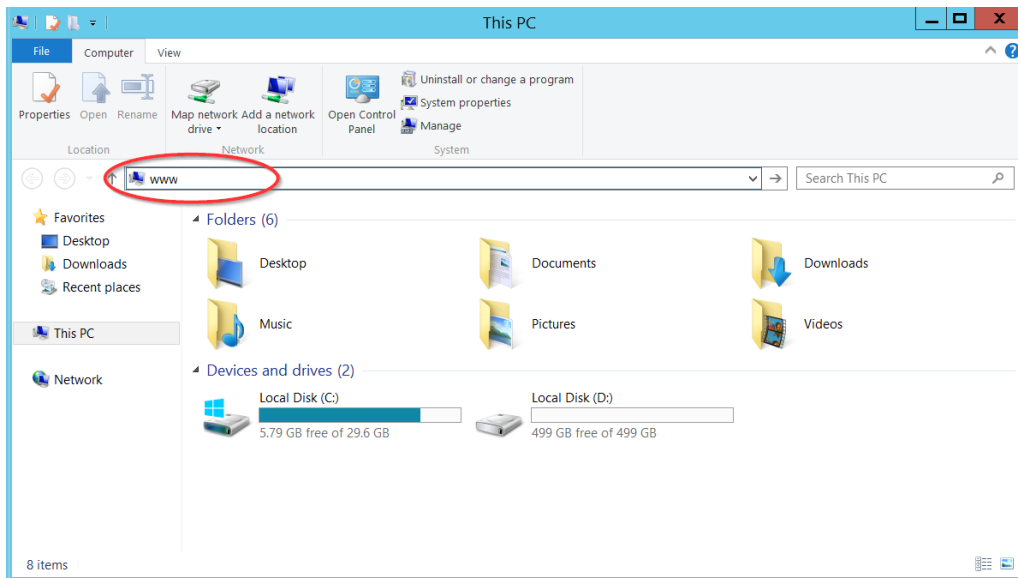


13. Click **Yes**.

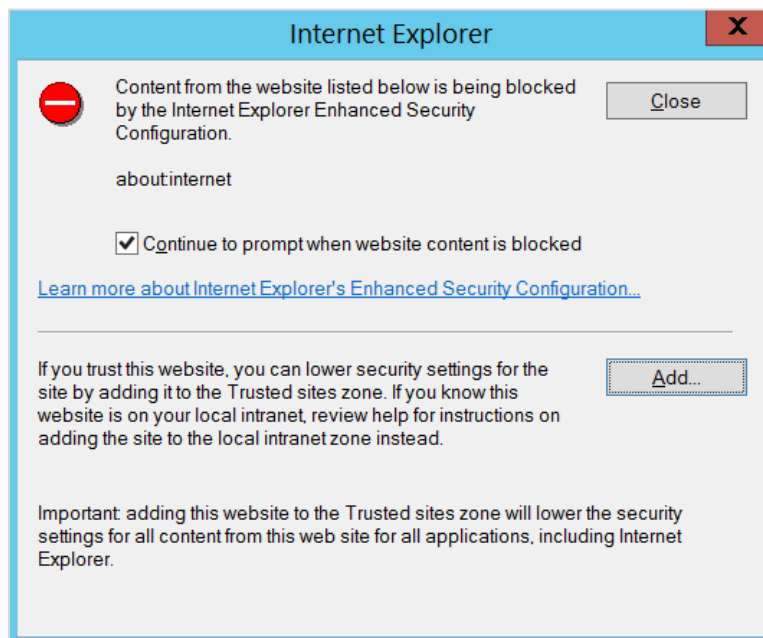
Your Remote Desktop session launches.



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

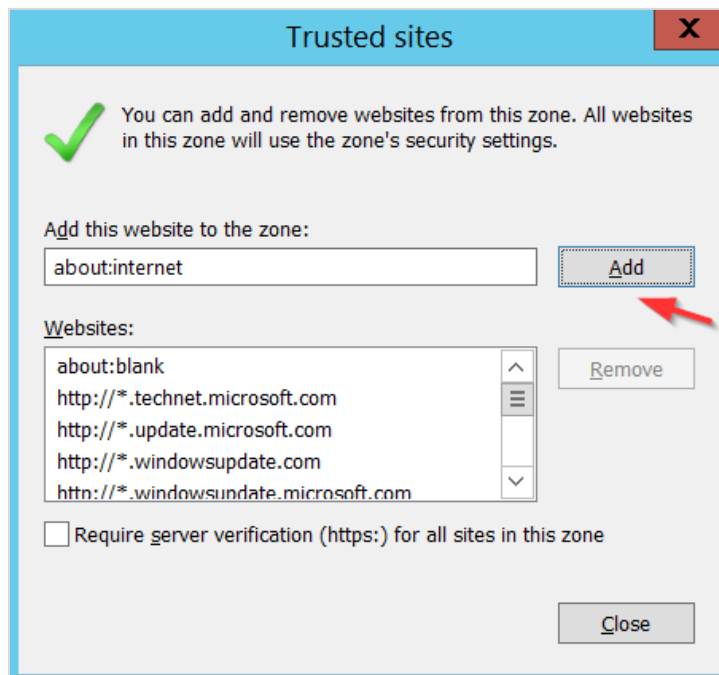


A message indicates that your browser is blocking you from reaching the internet.



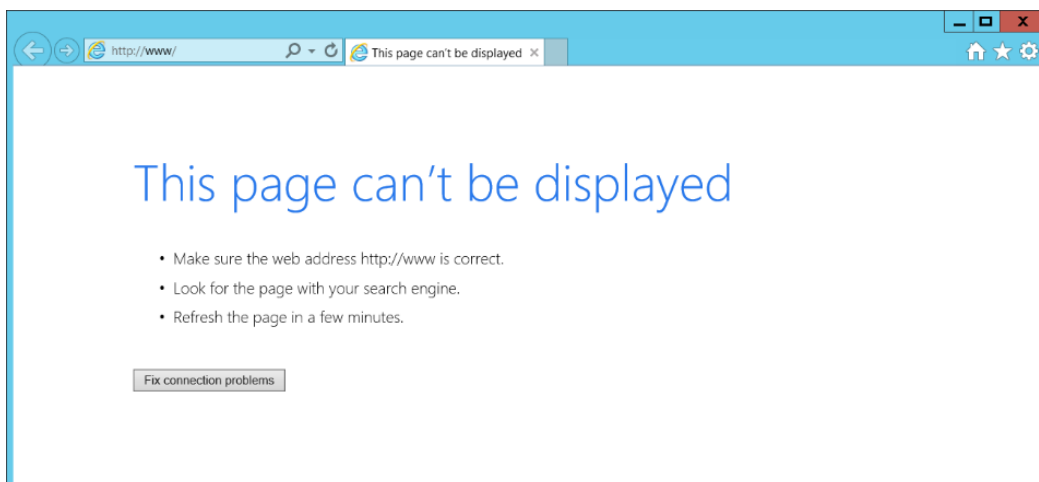
- Click **Add**. The Trusted sites dialog box opens.

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



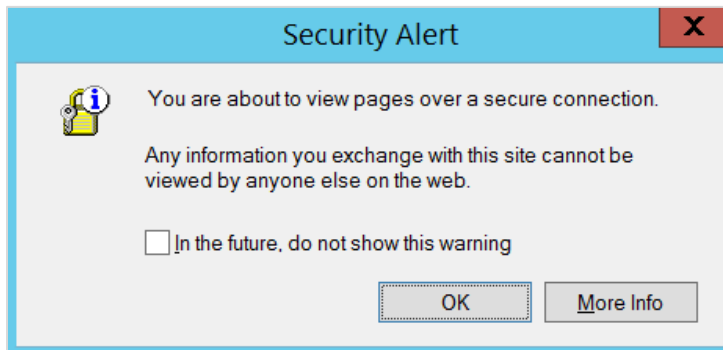
18. Click **Close**.

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



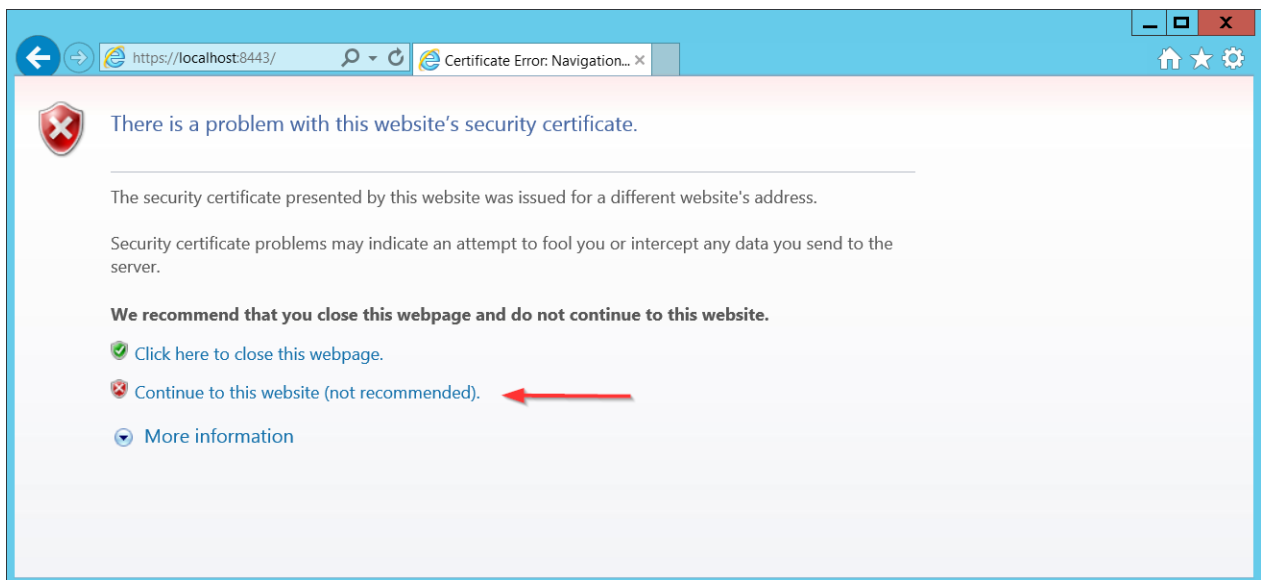
19. In the address bar, type **http://localhost:8080** and press **Enter**.

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



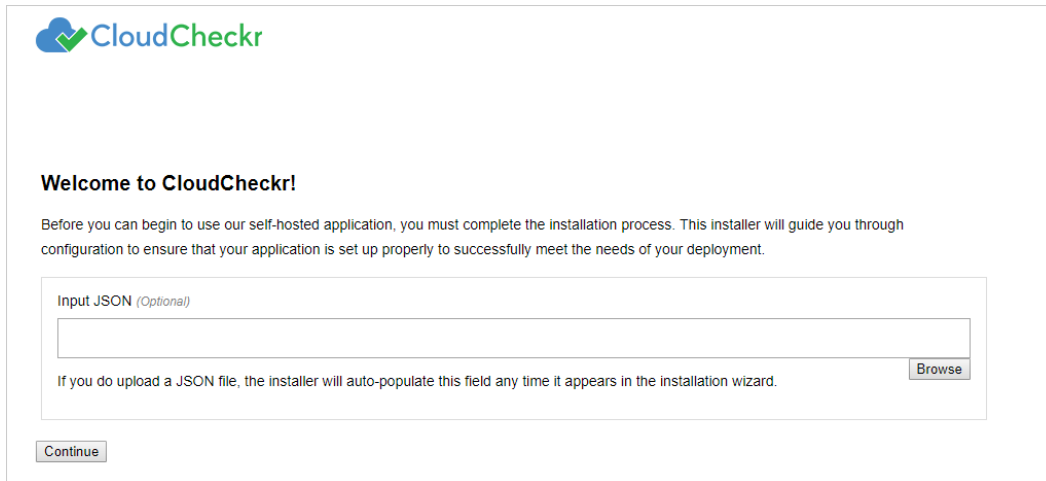
A warning about your security certificate displays.

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



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

The 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.



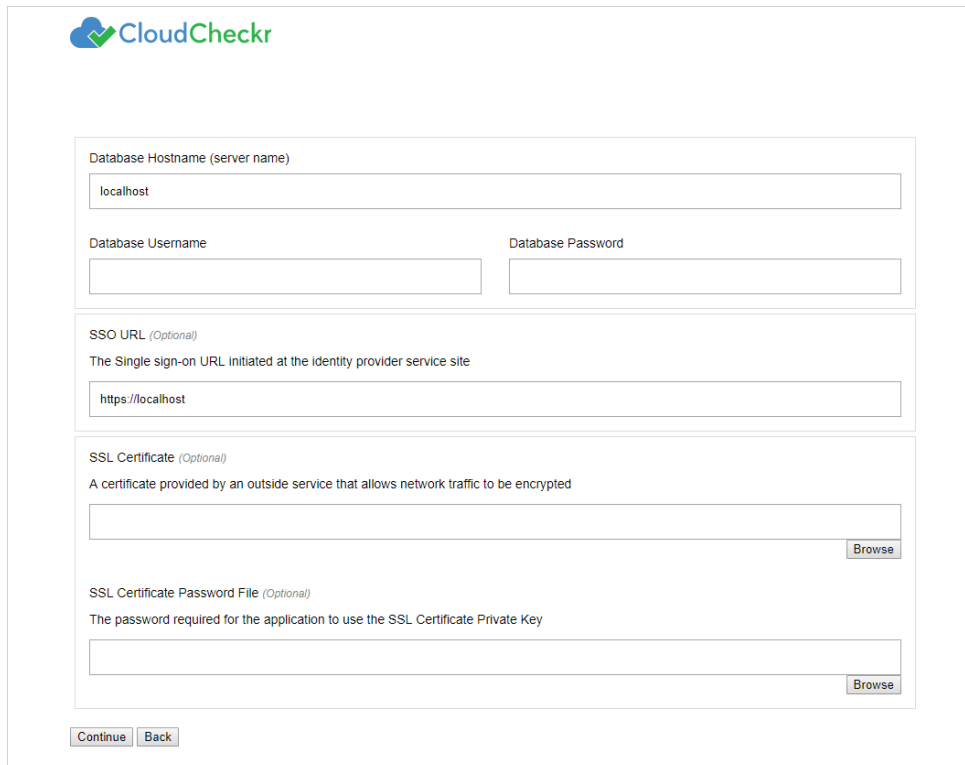
The screenshot shows the CloudCheckr logo at the top left. Below it, the heading "Welcome to CloudCheckr!" is followed by a paragraph: "Before you can begin to use our self-hosted application, you must complete the installation process. This installer will guide you through configuration to ensure that your application is set up properly to successfully meet the needs of your deployment." Below this is a form section titled "Input JSON (Optional)" containing a text input field and a "Browse" button. A note below the input field states: "If you do upload a JSON file, the installer will auto-populate this field any time it appears in the installation wizard." At the bottom left of the form is a "Continue" button.

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. Click **Browse** to navigate to the JSON file if you want to upload it. See the [Input JSON File](#) section.

24. Click **Continue**. The next 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. The form is divided into several sections:

- Database Hostname (server name):** A text input field containing "localhost".
- Database Username:** An empty text input field.
- Database Password:** An empty text input field.
- SSO URL (Optional):** A text input field containing "https://localhost". Below it is a descriptive text: "The Single sign-on URL initiated at the identity provider service site".
- SSL Certificate (Optional):** A text input field with a "Browse" button to its right. Below it is a descriptive text: "A certificate provided by an outside service that allows network traffic to be encrypted".
- SSL Certificate Password File (Optional):** A text input field with a "Browse" button to its right. Below it is a descriptive text: "The password required for the application to use the SSL Certificate Private Key".

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

25. 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

26. If applicable, provide the following information:

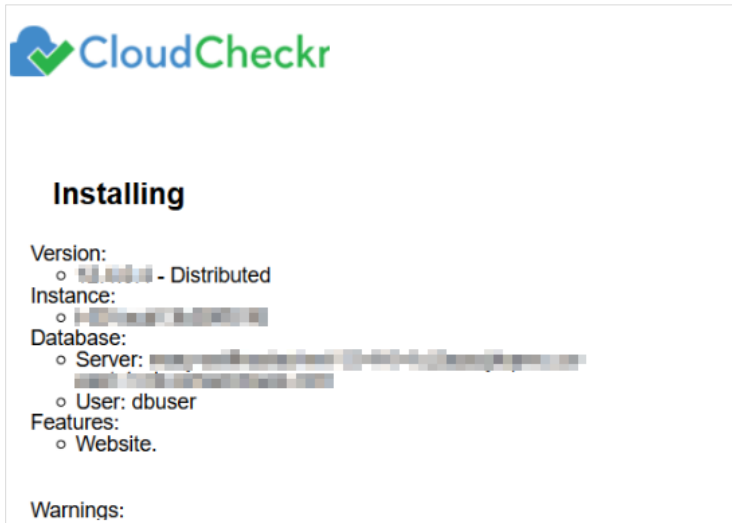
- **SSO URL:** Single Sign-On URL
- **SSL Certificate:** allows network traffic to be encrypted
- **SSL Certificate Password File:** password required for the application to use the SSL certificate private key

27. 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

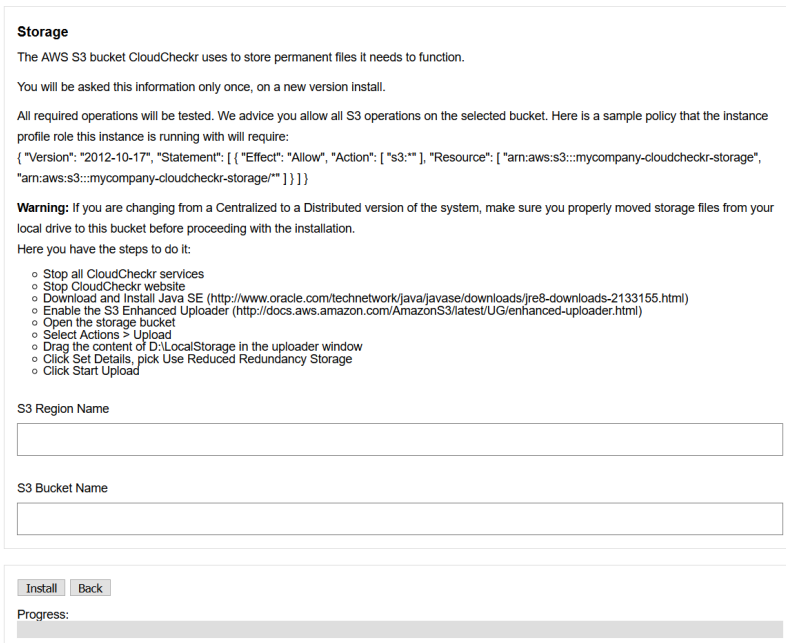
- verifies that the Microsoft SQL® server is available to communicate with the application
- identifies that the website (Web Console) is being installed



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



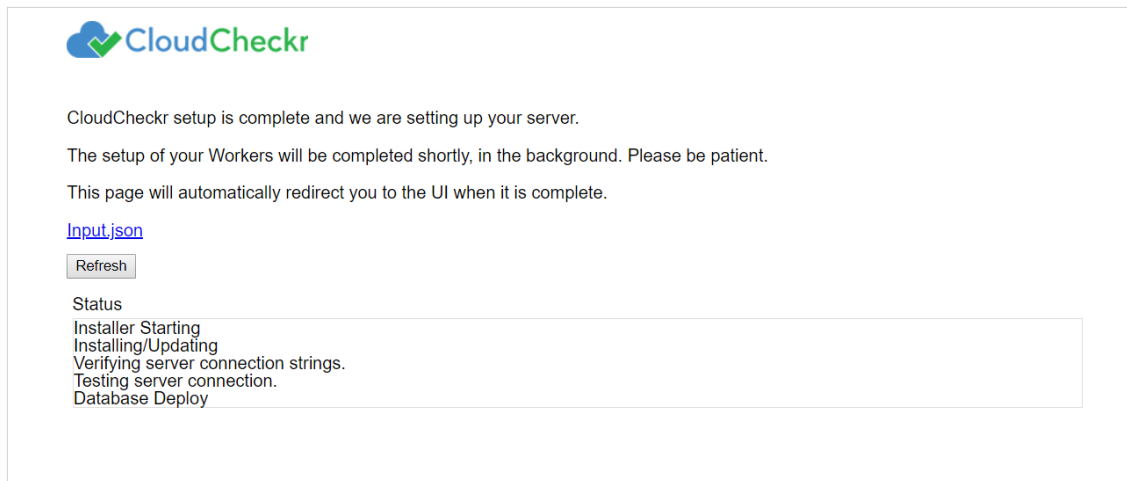
29. Scroll down to the Storage section. This is where you will identify the S3 region and bucket.



30. Copy the S3 region and bucket name from the [Required Information](#) section and paste them into the appropriate fields.

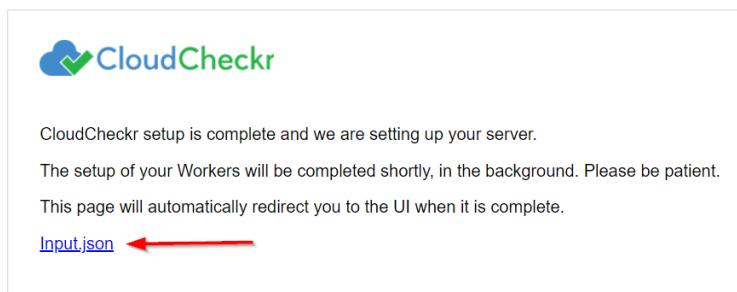
31. Click **Install**.

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:

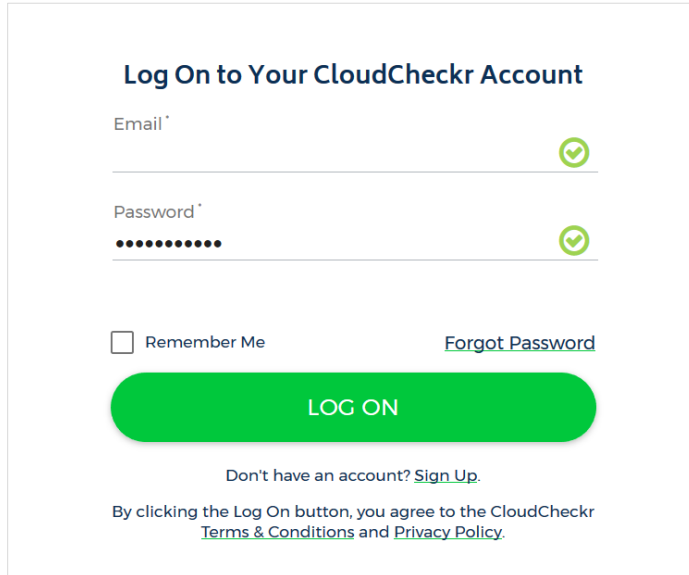


If you uploaded the file earlier, it will retain that same configuration. Since the filename is not important, you can rename the file to suit your needs. If you forget to click the link and you want to use the file later, go to:

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.



Log On to Your CloudCheckr Account

Email ✓

Password ✓

Remember Me [Forgot Password](#)

LOG ON

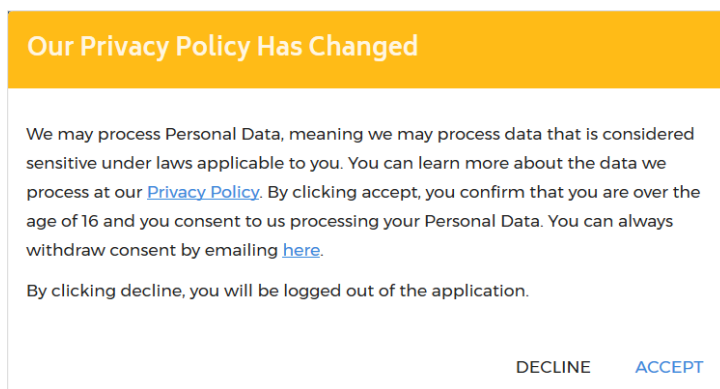
Don't have an account? [Sign Up](#).

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

32. In the Email text field, type **sysuser**
33. In the Password text field, paste the **EC2 instance ID** of the Web console.
34. Click **LOG ON**.

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

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



Our Privacy Policy Has Changed

We may process Personal Data, meaning we may process data that is considered sensitive under laws applicable to you. You can learn more about the data we process at our [Privacy Policy](#). By clicking accept, you confirm that you are over the age of 16 and you consent to us processing your Personal Data. You can always withdraw consent by emailing [here](#).

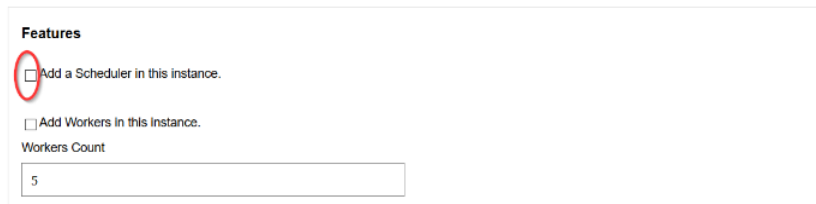
By clicking decline, you will be logged out of the application.

DECLINE **ACCEPT**

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



Features

Add a Scheduler in this instance.

Add Workers in this instance.

Workers Count

5

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**



Features

Add a Scheduler in this instance.

Add Workers in this instance.

Workers Count

5

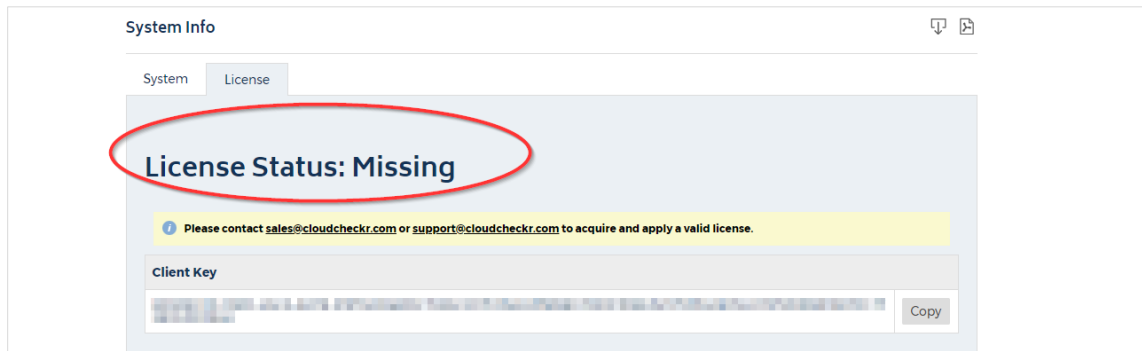
CONFIGURE THE SELF-HOSTED APP

License Your App

Note:

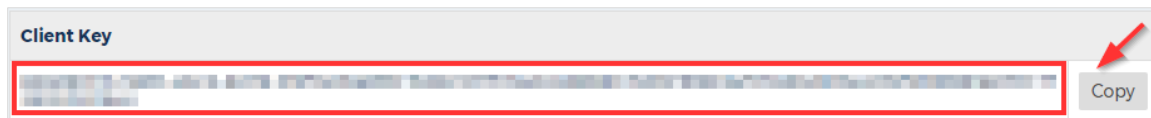
- If you purchased the AMI from AWS Marketplace, you are **not required** to license the self-hosted application. Skip this section and go to the [Create a Partner](#) section.
- If you purchased the AMI privately from CloudCheckr, you are **required** to license the self-hosted application for all versions 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**.



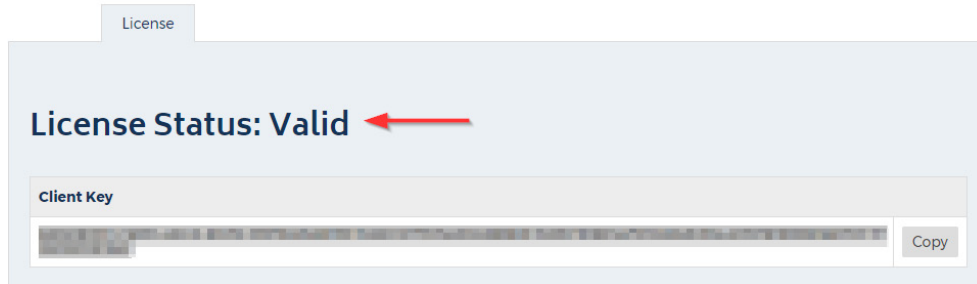
2. Email your sales representative to request a new license. Be sure to include your client key.

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.
 - o 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.
 - o 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**.

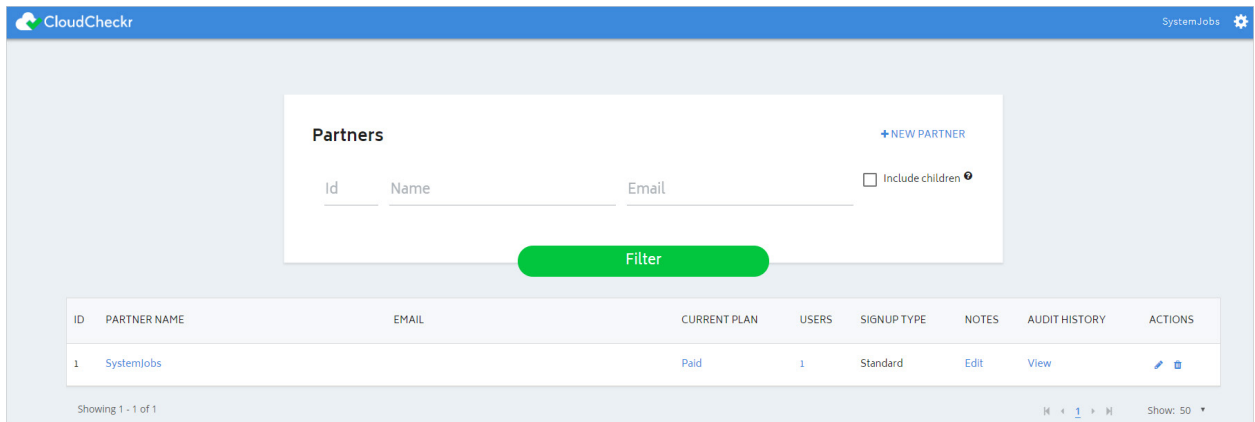
Once the application loads the license file, the License Status changes to **Valid**.



Create a Partner

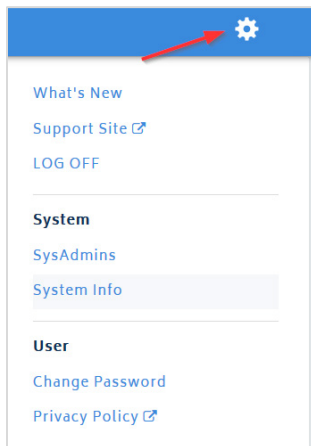
The first step in the configuration of your self-hosted application is to create a **partner**—the top-level container where you will store your accounts. More than likely, you will only need one partner—especially if you want all your accounts in one location.

The main landing page is the Partners page:



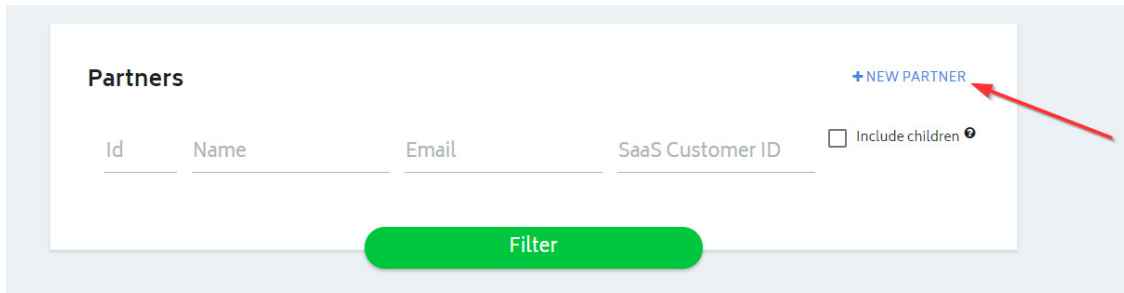
- If you just finished licensing your self-hosted application, click the **Back to Partners List** button in the License tab to return the Partners page.

Because you have not configured the application, you will only see the Settings icon 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**.



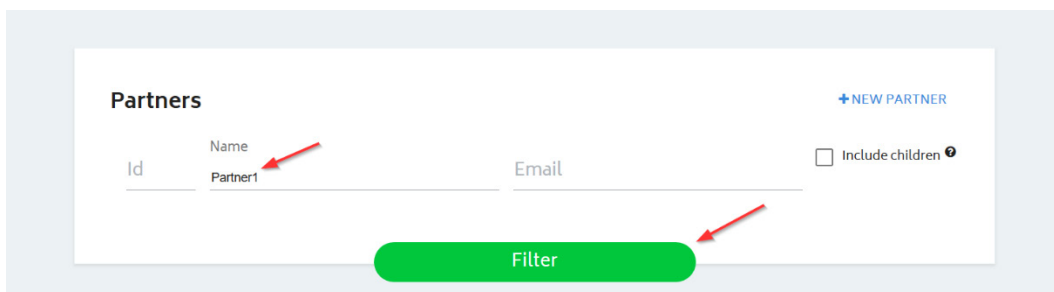
The Add Partner dialog box opens.

A screenshot of the "Add Partner" dialog box. It has a blue header with the text "Add Partner". Below the header is a section titled "Partner Information" with the instruction "Enter a name for your new partner. An email address is only required if an initial user is added." There are two text input fields: "Partner Name" and "Partner Email". Below this is a section titled "Initial User" with the instruction "If you choose to add a user to the partner, you can optionally set a password. If none is provided, the user will be required to set one on activation." There is a checkbox labeled "Add an initial user to the partner". At the bottom right of the dialog are two buttons: "CANCEL" and "CREATE".

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**.

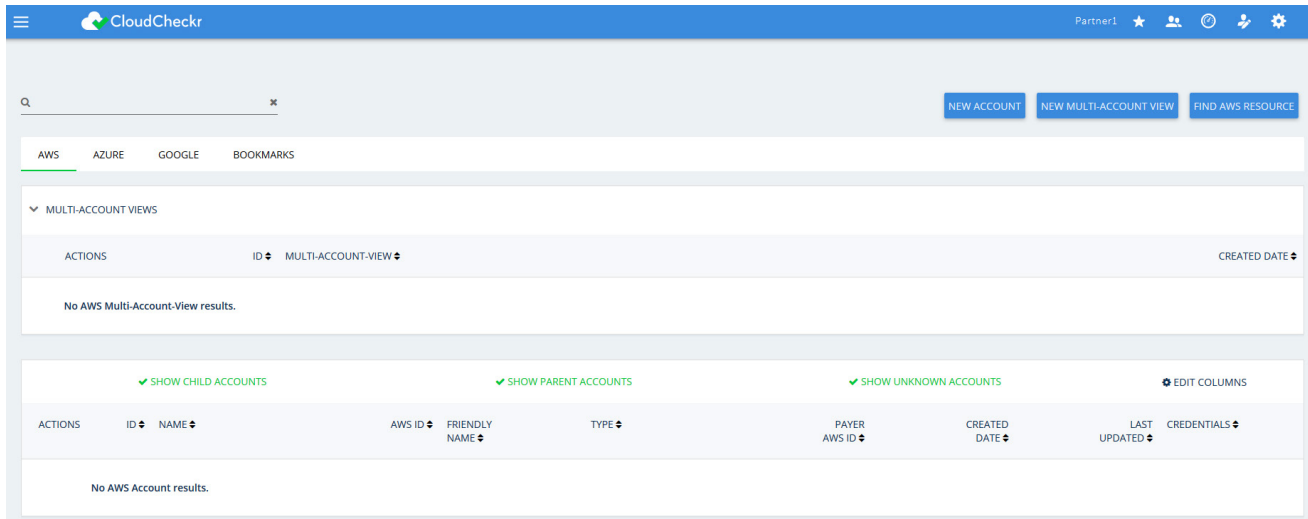
A message indicates that the application added the partner successfully.

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







CloudCheckr adds your new partner to the partner list.

- 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 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), 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.

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 and paste the values of the access and secret keys you created in the [Create an IAM User](#) 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

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.

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

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.

Session Tokens from the STS endpoints

AWS recommends using regional STS endpoints to reduce latency. Session tokens from regional STS endpoints are valid in all AWS Regions. If you use regional STS endpoints, no action is required.

Session tokens from the global STS endpoint (<https://sts.amazonaws.com>) are valid only in AWS Regions that are enabled by default. If you intend to enable a new Region for your account, you can use session tokens from regional STS endpoints or activate the global STS endpoint to issue session tokens that are valid in all AWS Regions. [Learn more](#)

Endpoints	Region compatibility of session tokens	Actions
Global endpoint	Valid in all AWS Regions	Edit
Regional endpoints	Valid in all AWS Regions	

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**.

Change region compatibility of session tokens for global endpoint ✕

Session tokens valid in all AWS Regions are larger. If you store session tokens, these larger tokens might affect your systems. [Learn more](#)

Session token's from the global endpoint (<https://sts.amazonaws.com>):

Only valid in AWS Regions enabled by default

Valid in all AWS Regions ←

Cancel Save changes

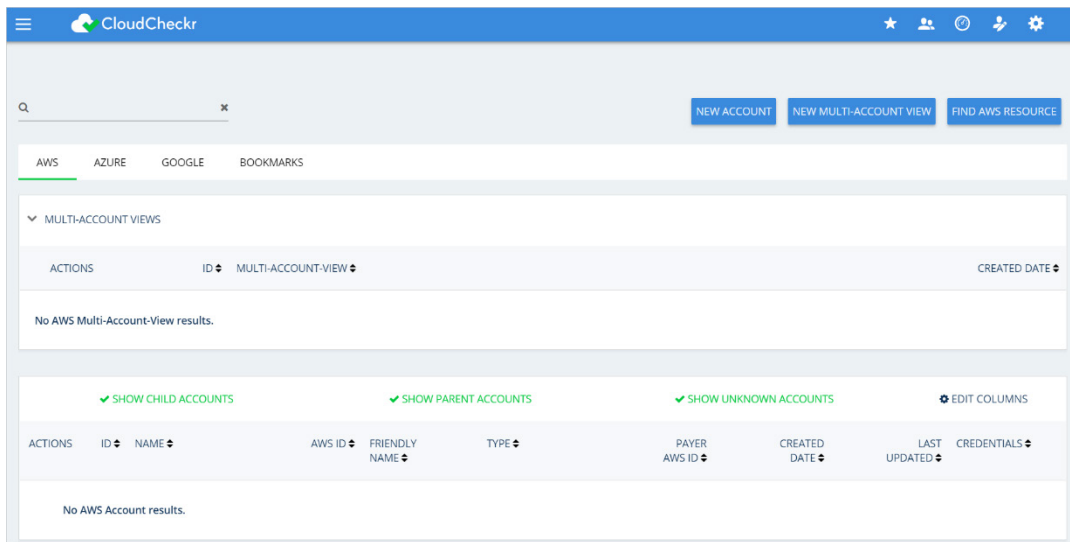
Create an Account

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**.

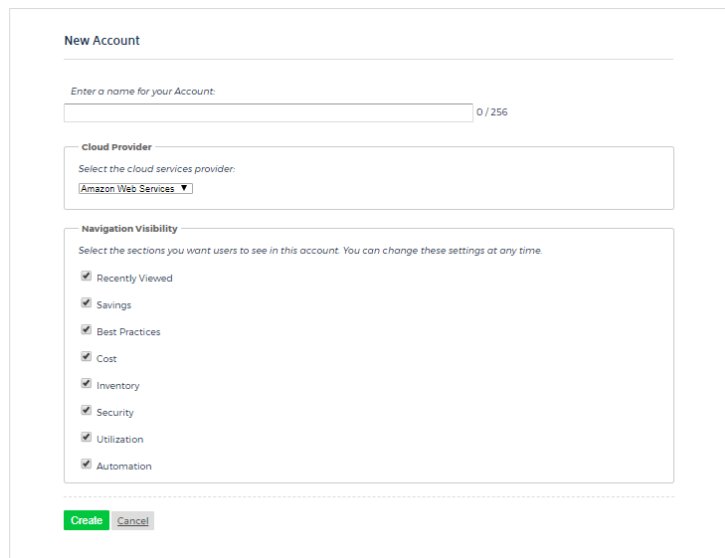
Note: You can also go to the Partners landing page and click **your partner name**.

The Accounts page for your partner displays.



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

The New Account screen displays.

The 'New Account' form is displayed. It has a title 'New Account' and a text input field for 'Enter a name for your Account' with a character count of '0 / 256'. Below this is a 'Cloud Provider' section with a dropdown menu currently set to 'Amazon Web Services'. The 'Navigation Visibility' section contains a list of checkboxes for various account sections: 'Recently Viewed', 'Savings', 'Best Practices', 'Cost', 'Inventory', 'Security', 'Utilization', and 'Automation'. All checkboxes are checked. At the bottom of the form are 'Create' and 'Cancel' buttons.

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](#).

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](#)

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](#).

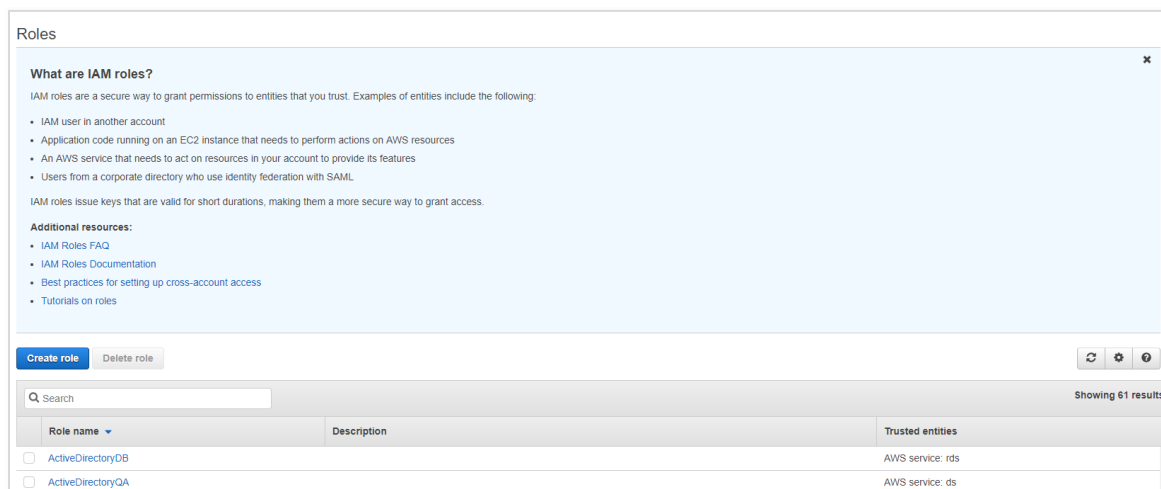
2. Copy the names of the least privilege policies to the [Required Information](#) section.

Create a Cross-Account Role

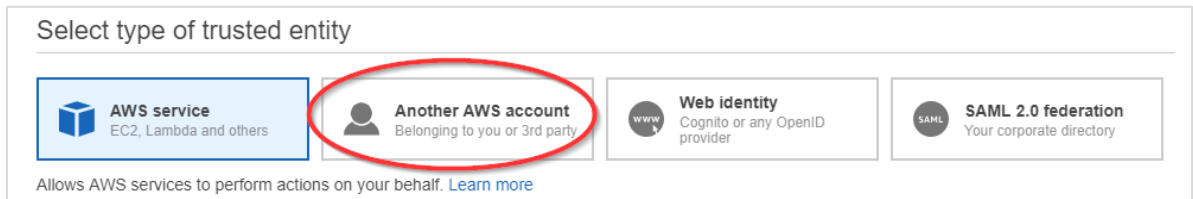
To finish your account configuration, you must create a cross-account role in AWS and apply those credentials in CloudCheckr.

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

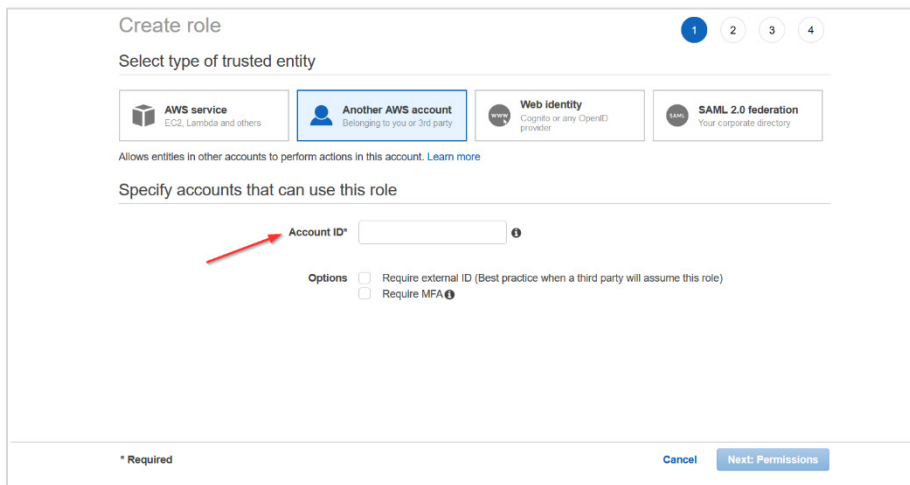
2. From the dashboard, click **Roles**. The Roles page opens.



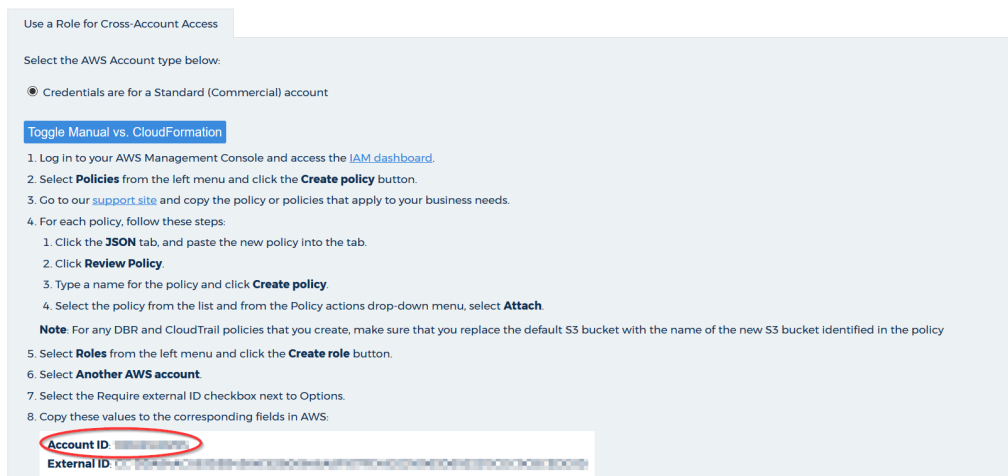
3. From the middle of the page, click **Create role**.
4. In the Select type of trusted entity section, click **Another AWS account**.



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.

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

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](#)

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.

Use a Role for Cross-Account Access

Select the AWS Account type below.

Credentials are for a Standard (Commercial) account

[Toggle Manual vs. CloudFormation](#)

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

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

Account ID

External ID

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.

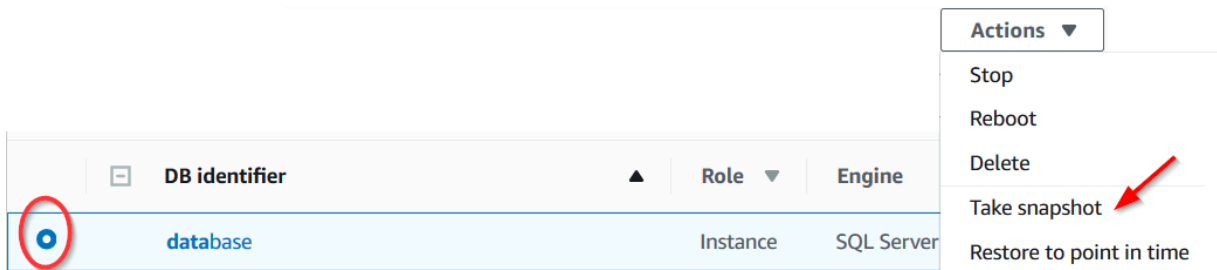
UPGRADE THE SELF-HOSTED APP

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

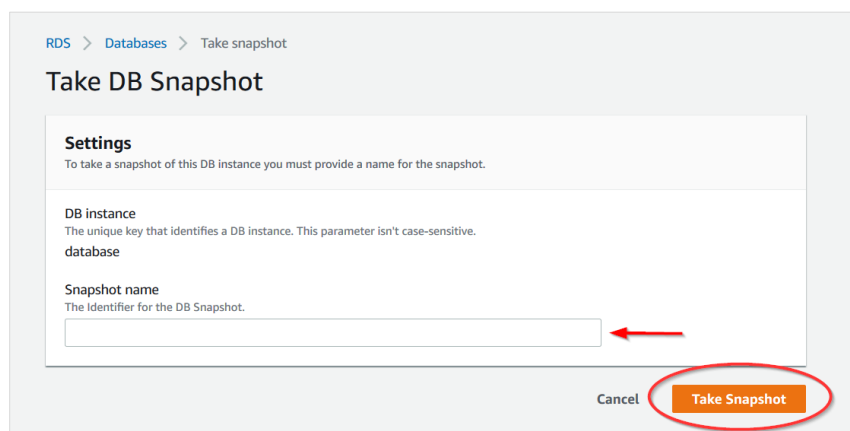
- create an RDS snapshot in AWS
- launch the new AMI
- install the new application
- license the new application

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** AMI.

- Complete the steps in the [Configure the EC2 Instances](#) section.
- 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:

CloudCheckr

Database Hostname (server name)
original RDS database name

Database Username Database Password

SSO URL (Optional)
The Single sign-on URL initiated at the identity provider service site
https://localhost

SSL Certificate (Optional)
A certificate provided by an outside service that allows network traffic to be encrypted
Browse

SSL Certificate Password File (Optional)
The password required for the application to use the SSL Certificate Private Key
Browse

Continue Back

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

CloudCheckr

<p>Installing</p> <p>Version: ◦ 15.4.0.4 - Centralhost</p> <p>Instance: ◦ H01110027-041-03-H02968</p> <p>Database: ◦ Server: localhost ◦ User: [Integrated Security]</p> <p>Features: ◦ Website. ◦ Scheduler service. ◦ Worker services.</p>	<p>Currently Running</p> <p>Version: ◦ 15.4.0.4 - Centralhost</p> <p>Website: ◦ Instance: I-01110027-041-03-H02968</p> <p>Scheduler: ◦ Instance: H01110027-041-03-H02968</p> <p>Workers: ◦ Instance: H01110027-041-03-H02968 Count: 5</p>
---	--

- If you purchased a private offering, follow the [License the App](#) section to upload the new license file.

FREQUENTLY ASKED QUESTIONS

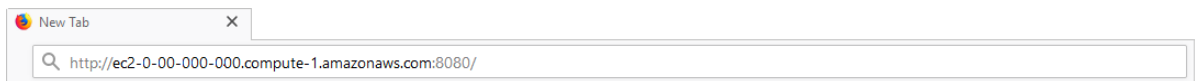
Is There an Alternative to Remote Desktop?

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.

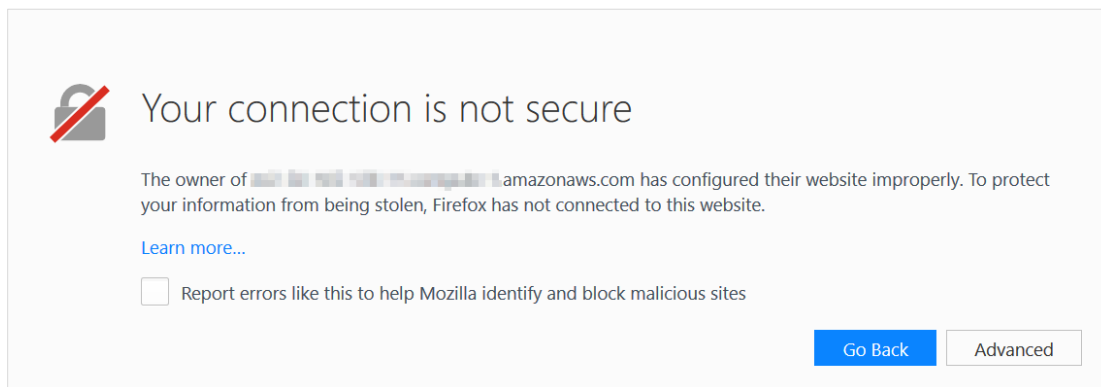
Description	
Instance ID	i-00d309a083e4f0435
Instance state	
Instance type	t3.small
Elastic IPs	0.00.000.000*
Public DNS (IPv4)	ec2-0-00-000-000.compute-1.amazonaws.com
IPv4 Public IP	0.00.000.000

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.

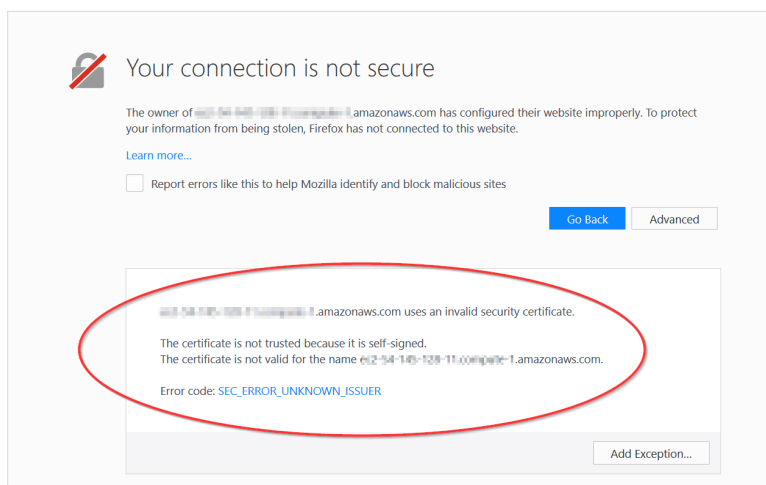
When the configuration is complete, a warning message indicates that your connection is not secure.



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

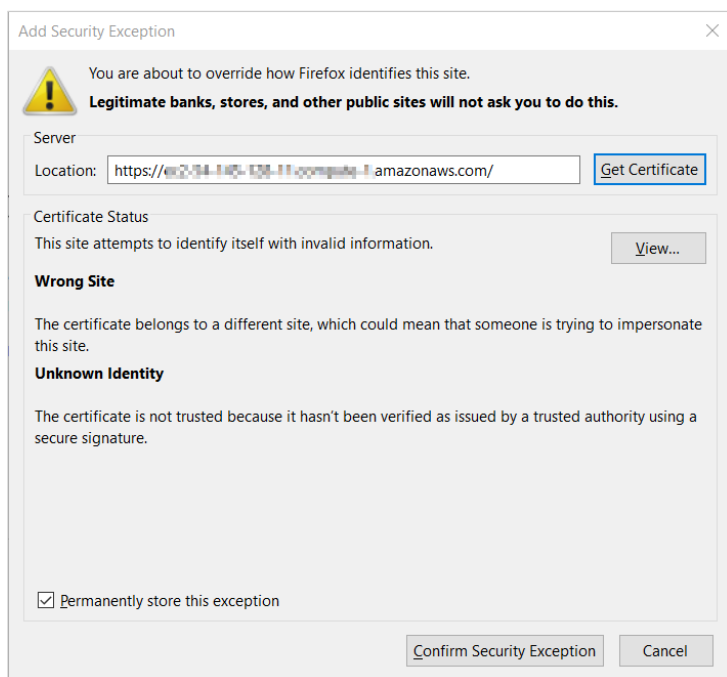
The application requires a secure connection with a certificate owned by the domain. Since you are launching the application in a self-hosted environment, it cannot automatically create a certificate.

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.

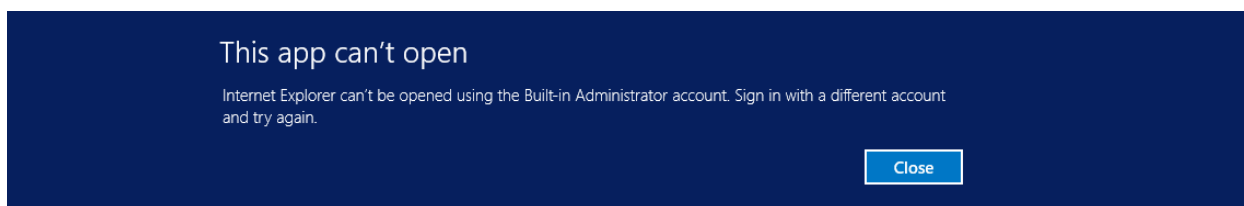
The Add Security Exception dialog box opens.



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?

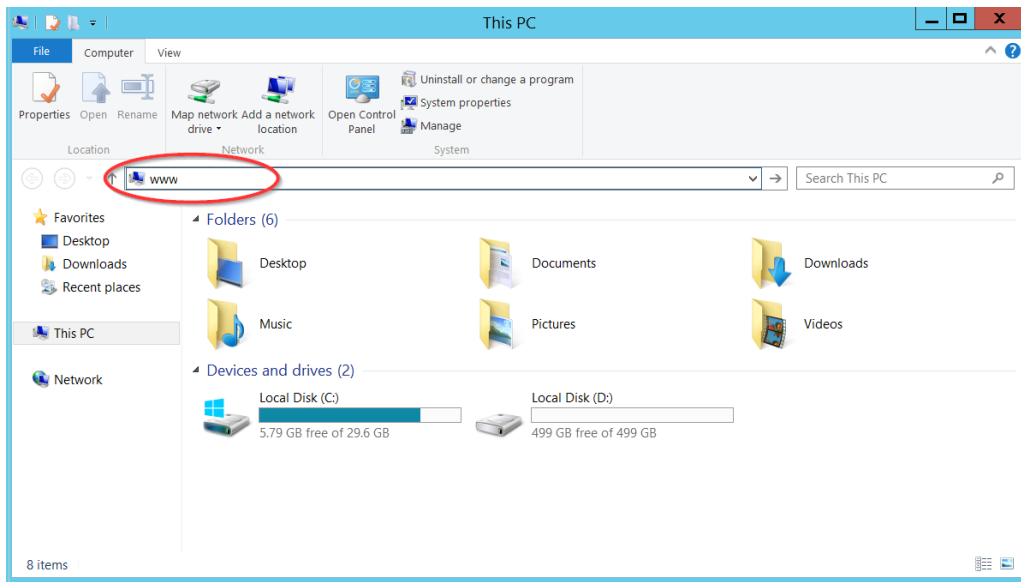
Your remote desktop session runs in a Microsoft Windows® 2012 R2 server environment, which is not compatible with newer applications like Internet Explorer. As a result, since you are logging in as an administrator, you will get an error message when you select **Start > Internet Explorer**:



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.

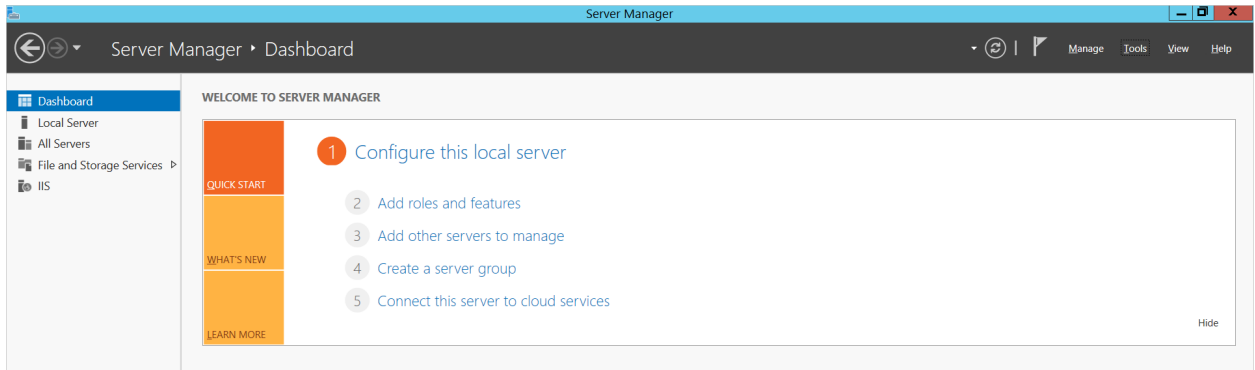
Where Is My D: Drive?

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.

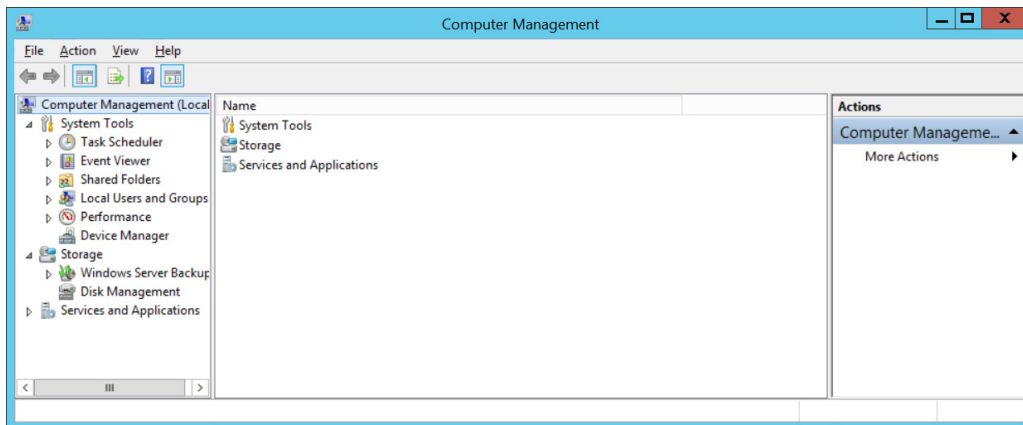


The Server Manager Dashboard opens.

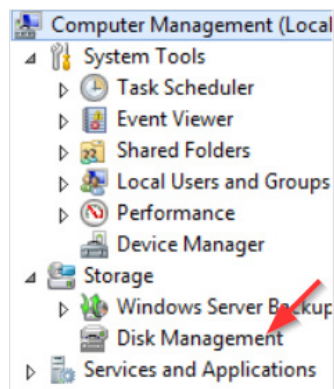


2. From the menu bar, choose **Tools > Computer Management**.

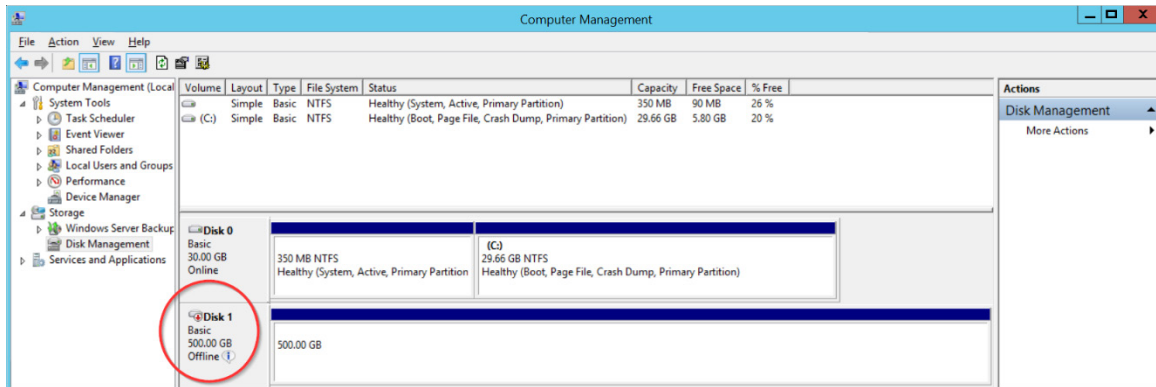
The Computer Management screen displays.



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**.

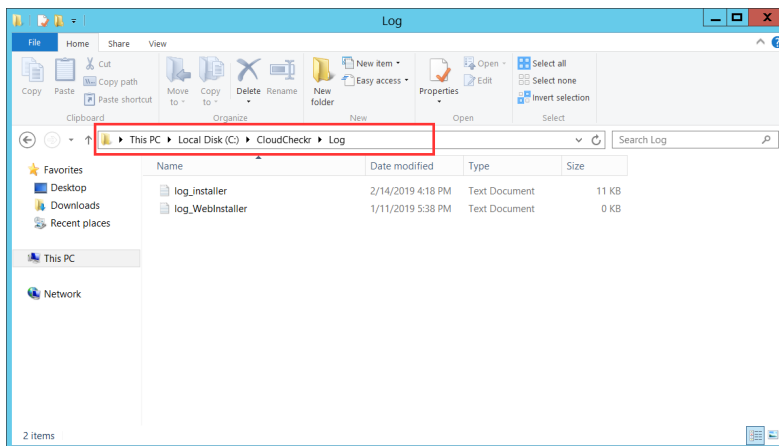


4. Right-click the **disk name** and from the fly-out menu, select **Online**. Your D: drive is now available.

How Do I Access My Log Files?

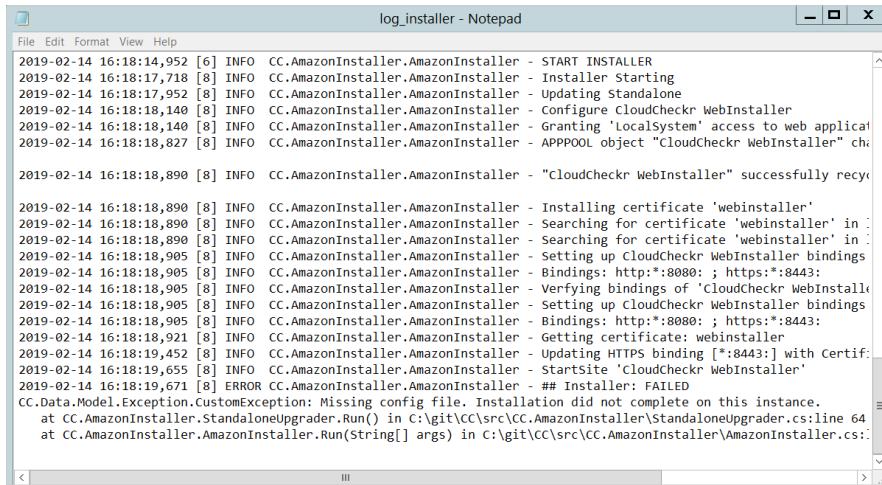
We can help you diagnose and solve the problem by reviewing your **log files**, which record every action performed within the web installer and the application. Follow these steps to access your log files:

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.



```
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.

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	
Private IP for SQL Server	
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]/*"
      ]
    }
  ]
}
```

```

{
  "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 (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 version of CloudCheckr is available in the event of an outage, we recommend that you install your Web host 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.