Console Guide

Amazon DCV Access Console

Amazon DCV Access Console: Console Guide

Table of Contents

What is Amazon DCV Access Console?

Note

Amazon DCV was previously known as NICE DCV.

The Amazon DCV Access Console is a web application that helps administrators and end users manage their Amazon DCV sessions. The Access Console consists of installable software packages that include a Handler, an Authentication Server, and a Web Client configured to provide a graphical interface.

The Access Console provides administrators with the following:

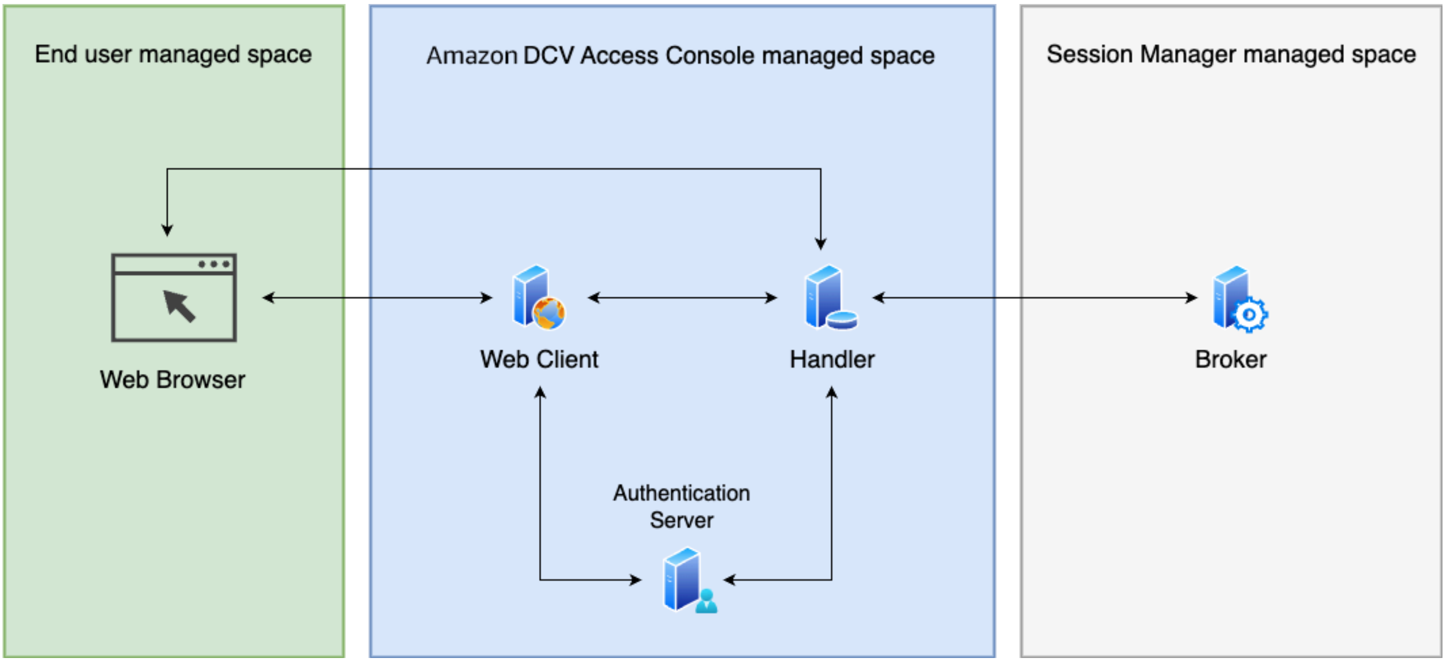
- Access to the Amazon DCV Session Manager APIs
- The ability to monitor the host servers running their sessions
- Tools to manage the users who have access to the console

The Access Console provides end users a way to connect, manage, and launch their own Amazon DCV sessions.

Topics

How Amazon DCV Access Console works

The following system architecture diagram shows the high-level components of the Amazon DCV Access Console and how they work with each other.



Handler

The *Handler* is an application that helps connect to and manage Amazon DCV sessions by communicating with the *Session Manager Broker* using the *Session Manager APIs*.

Authentication Server

The *Authentication Server* is responsible for authenticating users using Header based or PAM authentication methods.

Web Client

The client is the front-end web application you setup to interact with the *Handler* (and in turn with the *Session Manager Broker*). It renders the relevant web pages and serves to the *Web Browser*.

Session Manager Broker

The *Broker* is a web server that hosts and exposes the Session Manager APIs. It receives and processes *API* requests to manage Amazon DCV sessions from the *client*, and then passes the instructions to the relevant *Agents*. The Broker must be installed on a host that's separate from your Amazon DCV servers. It must also be accessible to the client, and be able to access the Agents.

Features

Amazon DCV Access Console offers the following features:

- **Provides Amazon DCV session information**–get information about the sessions running on multiple Amazon DCV servers.
- **Manage the lifecycle for multiple Amazon DCV sessions**–create or delete multiple sessions for multiple users across multiple Amazon DCV servers with one API request.
- **Supports tags**–use custom tags to target a group of Amazon DCV servers when creating sessions.
- **Manages permissions for multiple Amazon DCV sessions**–modify user permissions for multiple sessions with one API request.
- **Provides connection information**–retrieve client connection information for Amazon DCV sessions.
- **Supports for cloud and on-premises**–use Session Manager on AWS, on-premises, or with alternative cloud-based servers.

Limitations

Amazon DCV Access Console does not provide resource provisioning capabilities. If you are running Amazon DCV on Amazon EC2 instances, you might need to use additional AWS services, such as Amazon EC2 Auto Scaling to manage the scaling of your infrastructure.

Pricing

Amazon DCV Access Console is available at no cost for AWS customers running EC2 instances.

On-premises customers require a Amazon DCV Plus or Amazon DCV Professional Plus license. For information about how to purchase a Amazon DCV Plus or Amazon DCV Professional Plus license, see How to Buy on the Amazon DCV website. You can also use the website to find an Amazon DCV distributor or reseller in your region. Licensing requirements will only be enforced starting with Amazon DCV version 2021.0,so that all on-premises customers can experiment with the Amazon DCV Access Console.

For more information, see Licensing the Amazon DCV Server in the *Amazon DCV Administrator Guide*.

Requirements

The Amazon DCV Access Console has the following requirements.

	Authentication Server	Handler	Web Client
Operating system	- AL 2023 - CentOS Stream 9 - RHEL 9.x - Rocky Linux 8.5 or later - Rocky Linux 9.x - Ubuntu 22.04 - Ubuntu 24.04	- AL 2023 - CentOS Stream 9 - RHEL 9.x - Rocky Linux 8.5 or later - Rocky Linux 9.x - Ubuntu 22.04 - Ubuntu 24.04	- AL 2023 - CentOS Stream 9 - RHEL 9.x - Rocky Linux 8.5 or later - Rocky Linux 9.x - Ubuntu 22.04 - Ubuntu 24.04
Browser	N/A	N/A	Latest Chrome Browser
Architecture	- 64-bit x86 - 64-bit ARM	- 64-bit x86 - 64-bit ARM	- 64-bit x86 - 64-bit ARM
Memory	4 GB	4 GB	4 GB
Additional requirements	Java 17	Java 17, DynamoDB/ MariaDB/MySQL	Node 18, NGNIX

Authentication methods

The Authentication Server for the Amazon DCV Access Console can be setup to use either Pluggable Authentication Modules (PAM), HTTP Header authentication, or external OAuth providers. Utilizing PAM authentication allows you to inherit your existing Linux authentication model. HTTP Header authentication provides a customizable authentication mechanism to perform additional validation before the end user reaches the authentication server. External OAuth providers, such as AWS Cognito, allow you to leverage managed identity services for user authentication and management.

PAM authentication

The authentication server can be setup to use PAM authentication, it validates the username and the password using the PAM method of the operating system on the host running the authentication server.

Enabling PAM authentication

1. Connect to the host that is running the authentication server.
2. Open `/etc/dcv-access-console-auth-server/access-console-auth-server.properties` with your preferred editor.
3. Comment out or remove the `authentication-header-name` property to disable header based authentication if it is present.
4. Set the `pam-helper-path to the full path of the dcvpamhelper` that is installed as part of the authentication server. By default this is `/usr/share/dcv-access-console-auth-server/dcvpamhelper`.
5. Set the `pam-service-name` to the name of the file in `/etc/pam.d` that should be used to authenticate users.

   - To use the host's authentication for Redhat based operating systems, set the `pam-service-name` property to `system-auth`.
   - To use the host's authentication for Ubuntu/Debian based operating systems, set the `pam-service-name` to `common-auth`.
6. If the host uses different format of the username that are mapped to the same user in the operating system with the same uid and gid, set the `pam-normalize-userid-enabled` to true in order to normalize the username.

The userid is normalized using the command specified in `pam-normalize-userid-command`, by default it runs `id -u -nr` for each username and uses the output of the command as the userid.

7. Restart the authentication server.

```
sudo systemctl restart dcv-access-console-auth-server
```

HTTP Header authentication

The Amazon DCV Access Console can be setup to use the HTTP header in the request to the Authentication Server to authenticate a user. The Authentication Server checks for the configured header name in the request and uses the value of the header as the user id.

This method is useful when there is an intermediary identity provider between the Web Client and the Authentication Server. The intermediary solution authenticates the user and forwards the request with the configured HTTP header. For example, the authentication server can be setup behind a load balancer which uses an Amazon Incognito user pool to validate the user.

> **Note**
>
> It is important that the intermediary solution removes the configured header name from the requests from the web browser so that users cannot bypass the authentication solution.

Configuring HTTP header authentication

1. Connect to the host that is running the authentication server.
2. Open `/etc/dcv-session-manager-ui-auth-server/session-manager-auth-server.properties` with your preferred editor.
3. Disable PAM based authentication if it is present, by commenting out or removing the `pam-helper-path` property.
4. Set the `authentication-header-name` to the header name in the request and use the value of the header as the userid.
5. Restart the authentication server.

```
sudo systemctl restart dcv-access-console-auth-server
```

External authentication with AWS Cognito

The Amazon DCV Access Console can be configured to use external OAuth providers for authentication. The following shows how to configure AWS Cognito as an OAuth provider.

Setting up AWS Cognito for external oAuth

1. Go to AWS Cognito on the AWS Management Console > User pools > Create user pool
2. Set up resources for your application and Create user directory:
   - Define your application- Traditional web application
   - Configure options as you like
   - Add a return URL: `<web-client-url>/api/auth/callback/<NEXT_PUBLIC_SM_UI_AUTH_ID>`. For example, using defaults for a locally running server: `http://localhost:3000/api/auth/callback/dcv-access-console-auth-server`
   - Once the user pool is created, you can configure Allowed sign-out URLs: Applications > App clients > Login pages > Managed login pages configuration > Edit
3. Adding users to the user pool:
   - Go to User management > Users and add users
   - Alternatively, if you have allowed self-registration in step 2, users may sign up themselves
4. Preparing access-console-handler.properties:
   - Copy the User pool ID from the user pool Overview page and set `jwt-issuer-uri` as `https://cognito-idp.<region>.amazonaws.com/<user_pool_id>`
   - Set the following properties:
     - `jwt-login-username-claim-key` is the key for the login username claim key
     - `jwt-display-name-claim-key` is the key for the display name claim key
     - `auth-server-well-known-uri` is the well known URI (required only if userInfo endpoint is not provided) in the format `https://cognito-idp.<region>.amazonaws.com/<user_pool_id>/.well-known/openid-configuration`

    - `auth-server-userinfo-endpoint` is the userInfo endpoint
  - Restart the handler: `sudo systemctl restart dcv-access-console-handler`
  - Confirm that the service is running: `sudo systemctl status dcv-access-console-handler`
  - To get service logs: `sudo journalctl -u dcv-access-console-handler`

5. Preparing the web client:

   - `/etc/dcv-access-console-web-client/access-console-web-client.properties`:
     - Set `auth-server-well-known-uri` in the format `https://cognito-idp.<region>.amazonaws.com/<user_pool_id>/.well-known/openid-configuration`
   - `/etc/dcv-access-console-web-client/access-console-web-client-secrets.properties`:
     - Set the `auth-server-client-id` and `auth-server-client-secret` values as the Client ID and Client secret values of the user pool App client you set up in step 2 above (Applications > App clients > Select your App client name > App client information)
   - Restart the web client: `sudo systemctl restart dcv-access-console-web-client`
   - Confirm that the service is running: `sudo systemctl status dcv-access-console-web-client`
   - To get service logs: `sudo journalctl -u dcv-access-console-web-client`

Datastore

Amazon DCV Access Console persists user data, group data, session templates and the permission data related to them through integrations with external databases. It supports DynamoDB, MariaDB, and MySQL databases. You must set up and manage one of these databases to use Amazon DCV Access Console. If your Amazon DCV Access Console machines are hosted on Amazon EC2, we recommend using DynamoDB as the external database, since it does not require any additional setup.

Note

Additional costs can happen when running an external database. To see information on DynamoDB pricing, see Pricing for Provisioned Capacity.

Configure the Amazon DCV Access Console to persist on DynamoDB

1. On the host running the Handler component, open `/etc/dcv-access-console-handler/access-console-handler.properties` in your preferred editor and make the following edits:

   - Set `datastore = dynamodb`.
   - For `dynamodb-region` specify the AWS Region where you want to store the tables containing the Handler component data. For the list of supported Regions, see DynamoDB service endpoints.
   - For `datastore.prefix` specify the prefix that is added to each DynamoDB table (useful to distinguish multiple Handler component using the same account). Only alphanumeric characters, dot, dash, and underscore are allowed.

2. Stop the Handler component.

   ```
   sudo systemctl stop dcv-access-console-handler
   ```

3. Start the Handler component.

   ```
   sudo systemctl start dcv-access-console-handler
   ```

   The Handler component host must have permission to call the DynamoDB APIs. On Amazon EC2 instances, the credentials are automatically retrieved using the Amazon EC2 metadata service. If you need to specify different credentials, you can set them using one of the supported credential retrieval techniques (such as Java system properties or environment variables). For more information, see Supplying and Retrieving AWS Credentials.

Configure the broker to persist on MariaDB/MySQL

1. On the host running the Handler component, open `/etc/dcv-access-console-handler/access-console-handler.properties` in your preferred editor and make the following edits:

   - Set `datastore = mysql`.
   - Set `jdbc-connection-url = jdbc:mysql://`*`db_endpoint`*`:`*`db_port`*`/`*`db_name`*

     In this configuration, *`db_endpoint`* is the database endpoint, *`db_port`* is the database port, and *`db_name`* is the database name.
   - For `datastore.prefix` specify the prefix that is added to each DynamoDB table (useful to distinguish multiple Handler component using the same account). Only alphanumeric characters, dot, dash, and underscore are allowed.

2. On the host running the Handler component, open `/etc/dcv-access-console-handler/access-console-handler-secrets.properties` in your preferred editor and make the following edits:

   - For `jdbc-user` specify the name of the user that has access to the database.
   - For `jdbc-password` specify the password of the user that has access to the database.

3. Stop the Handler component.

   ```
   sudo systemctl stop dcv-access-console-handler
   ```

4. Start the Handler component.

   ```
   sudo systemctl start dcv-access-console-handler
   ```

   Note

   The `/etc/dcv-access-console-handler/access-console-handler-secrets.properties` file contains sensitive data. By default, its write access is restricted to root and its read access is restricted to root and to the user running the Handler component. By default, this is the `dcvaccessconsole` user.

Certificates

In order to provide a HTTPS connection between the different components, a SSL certificate is required for each of the hosts. Customers are recommend to use their own manager certificates on each of the host. For non-production workloads, a self-signed SSL certificate can be used. For more information on creating a self-signed cert see Generating a self-signed certificate.

See instructions below on how to configure the different Amazon DCV Access Console components to use certificates.

Authentication Server

1. Connect to the host that is running the Authentication Server.
2. Open `/etc/dcv-access-console-auth-server/access-console-auth-server-secrets.properties` with your preferred editor and update the following properties:
    - `server.ssl.key-store-type` – Set to PKCS12.
    - `server.ssl.key-store` – Set to path of the JKS keystore.
    - `server.ssl.enabled` – Set to true.
    - `server.ssl.key-store-password` – Set to key store password.
3. Restart the Authentication Server service.

```
sudo systemctl restart dcv-access-console-auth-server
```

Handler

1. Connect to the host that is running the Handler
2. Open `/etc/dcv-access-console-handler/access-console-handler-secrets.properties` with your preferred editor and update the following properties:
    - `server.ssl.key-store-type` – Set to PKCS12.
    - `server.ssl.key-store` – Set to path of the JKS key store.
    - `server.ssl.enabled` – Set to true.
    - `server.ssl.key-store-password` – Set to key store password.
3. Restart the Handler service.

```
sudo systemctl restart dcv-access-console-handler
```

Web Client/NGNIX

1. Connect to the host that is running NGNIX.
2. Open `/etc/nginx/conf.d/dcv-access-console.conf` with your preferred editor and update the following properties:
   - `ssl_certificate` – Set to path to the certificate for the host.
   - `ssl_certificate_key` – Set to path to the key for the certificate.
3. Restart the NGNIX service.

   ```
   sudo systemctl restart ngnix
   ```

Networking and connectivity

The Amazon DCV Access Console components can all be installed on a single host or on different hosts.

Single host setup

In a single host setup, the Authentication Server, the Handler component and the Web Client are all installed on a single host. An NGINX server can be used to proxy requests from the web browser to the appropriate component. The web browser should be able to initiate secure, persistent, bi-directional HTTPS connections with NGNIX. All the components need bi-directional HTTP connection between each other on the configured port (see table below). In addition, the Handler component needs to be able to initiate secure, persistent, bi-directional HTTPS connections with the Broker and the persistence store (DynamoDB or MariabDB/MySQL).

Component	Default Port
Authentication Server	3000
Handler	8080

Component	Default Port
Web Client	9000

Multiple host setup

In multiple host setup, the Authentication Server, the Handler component and the Web Client can be all installed on different servers. An NGNIX server can be used to proxy requests from the web browser to the Web Client and establish a HTTPS between them. The Authentication Server and the Handler can be configured to accept HTTPS connections. All the components need bi-directional HTTPs connection between them on port 443. In addition, the Handler component needs to be able to initiate secure, persistent, bi-directional HTTPs connections with the Broker and the persistence store (DynamoDB or MariabDB/MySQL).

Open source code

The Amazon DCV Access Console consists of installable software packages that include a Handler, an Authentication Server, a Web Client, and a Setup Wizard configured to provide a graphical interface for the Amazon DCV Session Manager broker. The Access Console is available as a packaged commercial build on the Amazon DCV downloads page and as separate open sourced components on GitHub. You may consider using these open source components if you want to customize the Access Console in ways not available with the commercial build to meet your unique use cases.

Other Amazon DCV products listed and available on the DCV downloads page, like the Amazon DCV Session Manager and the Amazon DCV clients, are not open sourced. If you choose to customize the Access Console using the open sourced Access Console components, the customized Access Console may be used in combination with the other DCV products as described below in the Licensing section.

Licensing

The Access Console code repositories stored on GitHub are open source and licensed under the Apache 2.0 License. The Access Console commercial build and other Amazon DCV products listed and available on our Amazon DCV downloads page are proprietary and licensed under the DCV EULA. If you use the open sourced Access Console components, but not the proprietary Amazon

DCV products, in a custom Access Console build, the customized Access Console is governed by the Apache 2.0 License. If you use the open sourced Access Console components in any combination that includes the proprietary Amazon DCV products (i.e., Amazon DCV Session Manager, Amazon DCV Clients, or other product listed on the Amazon DCV downloads page), the combination is governed by the DCV EULA.

Contributing

As a customer, you have the ability to contribute back to the open source repository. Follow the CONTRIBUTING instructions available on the GitHub repository for further instruction.

Prerequisites

Before setting up the Amazon DCV Access Console Access Manager, you must first install and configure the Session Manager Agent and Broker. For more information about setting up Amazon DCV Session Manager, see the Amazon DCV Session Manager Administrator Guide.

Registering a new client with the Broker

The Access Console has three components, the Web Client, the Handler, and the Authentication Server. You can set up the Access Console by:

- Running the Access Console components on the same host as the Session Manager Broker
- Running the Access Console components on a different host. If you choose this option, you must register a new client with the Broker. Use the following steps to register a new client with the Broker.

To register a new client with the Broker

1. Connect to the host where you installed the Broker.
2. Run the following command to register a new client:

   ```
   $ sudo -u root dcv-session-manager-broker register-api-client --client-name
    "access-console"
   ```

3. Take note of the `client-id` and `client-password`. We will need these when we set up the components.
4. The Broker host will also need to have a Public DNS assigned to it. Take note of the address. The Access Console Handler will need this to communicate with the Broker
5. Make sure that the host the Broker is running on is accessible by the host the Access Console Handler will be installed on, via the Broker's `client-to-broker-connector-https-port` and the Public DNS

   > **Note**
   >
   > If you haven't changed the default, this is port 8443

If the Broker is already running on the same host where you are going to install all three components, you don’t have to do anything. The Setup Wizard will register a new client with the broker for you.

Setting up Amazon DCV Access Console

When setting up your Amazon DCV Access Console, you can choose whether you want to run the console on a single host or, if you choose, across a set of multiple hosts. Using multiple hosts can improve scalability and performance. The Console works with either configuration.

The following section explains how to set up Amazon DCV Access Console on a single host and on separate multiple hosts.

Topics

Using the Setup Wizard

The Setup Wizard is a CLI designed to help you install the Amazon DCV Access Console, and configure the hosts you plan to install the components on. The Setup Wizard can be used whether you install the Access Console components all on the same host, or on separate hosts. If you install the components on a single host, it will install the components and dependencies for the Access Console for you. If you install the components on separate hosts, the Setup Wizard will help you create the configuration files needed for each component. The Setup Wizard can optionally:

- Install MariaDB using the OS package manager to act as a datastore. If you choose to use Amazon DynamoDB, no additional packages need to be installed.
- Create the necessary database in your chosen datastore
- Install NGINX using the OS package manager
- Generate and saves a self-signed certificate
- Install the Amazon DCV Access Console components
- Configure the Authentication Server with PAM authentication
- Start the datastore, NGNIX and the Amazon DCV Access Console components
- Create a user with the Admin role

- Validate that each component started correctly

> **Note**
>
> Through the Setup Wizard you may install certain third-party software that you can use in conjunction with the Amazon DCV Access Console. You are solely responsible for complying with any applicable terms and conditions for use of such third-party software, including obtaining any required licenses from the relevant third parties to use their technology and paying any necessary royalties or fees.

Running the wizard

The Setup Wizard in the Amazon DCV Access Console packaged components, available on Amazon DCV Downloads.

You can use the Setup Wizard in interactive or non-interactive mode to complete the setup of the Amazon DCV Access Console. The Setup Wizard will finish by validating the installation was successful then print the public DNS of the host you provided. The Amazon DCV Access Console will be accessible at that address and any user present on that host will be able to login.

Interactive mode

By default, the Setup Wizard runs in interactive mode. This mode prompts you to complete the required inputs. Run the Setup Wizard (see Run the Setup Wizard documentation for more details), and answer each prompt with the necessary requirements to setup the Amazon DCV Access Console.

Non-interactive mode

You can also choose to run the Setup Wizard in non-interactive mode. Using this mode, you manually fill in either the `onebox_wizard_input.json` or `wizard_input.json` file that comes with it or by using command-line options. The instructions for non-interactive mode are different, whether you install the Amazon DCV Access Console components on one host, or separate hosts.

Modifying setup wizard parameters

When in non-interactive mode, the Setup Wizard supports several ways of inputting parameter values.

Loading a JSON file

You can specify the input parameters by loading a JSON file to the Setup Wizard, where the key-value pairs are the name of the parameter and specified value. Two starter files are provided with the Setup Wizard: `wizard_input.json` for setting up on multiple hosts and `onebox_wizard_input.json` for setting up on a single host.

Example

For example, this file specifies the `broker-client-id` and the `broker-client-password`:

```
{
    "broker-client-id": "client_id"
    "broker-client-password": "client_password"
}
```

Then load the file into the Setup Wizard by specifying its path (absolute or relative) with the `--input-json` option. The Setup Wizard will prompt for any parameter not specified in the JSON file, unless the `--quiet` flag is used.

For a full list of the available options and flags, navigate to the folder where you extracted the Amazon DCV Access Console components:

```
$ python3 wizard.py --help
```

Command-line options

You can also specify the input parameters by using command-line options, for example `--broker-address`.

For a full list of the available options and flags, navigate to the folder where you extracted the Amazon DCV Access Console components and invoke:

```
$ python3 wizard.py --help
```

Setting up on a single host

This section explains how to install the Amazon DCV Access Console components on a single host. Before proceeding, you must first ensure you have completed the necessary Prerequisites.

To set up the Amazon DCV Access Console on a single host, do the following:

Steps

- Step 1: Prepare the environment
- Step 2: Run the Setup Wizard

Step 1: Prepare the environment

The Amazon DCV Access Console has three components Handler, Web Client, and Authentication Server. To streamline the setup process, you can install the components on the same host. See Amazon DCV Access Console Requirements to ensure your setup meets the requirements for setup on a single host.

Preparing the components and the Setup Wizard

1. Connect to the host on which you intend to install the Amazon DCV Access Console components.
2. Create a directory where you will save the installation files.

   ```
   $ mkdir dcv-access-console
   ```

   ```
   $ cd dcv-access-console
   ```

3. The Amazon DCV Access Console packages are digitally signed with a secure GPG signature. To allow the package manager to verify the package signature, you must import the Amazon DCV GPG key. To do so, open a terminal window and import the Amazon DCV GPG key by entering:

   - For all Linux distributions except Ubuntu:

     ```
     $ sudo rpm --import https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
     ```

   - For Ubuntu:

     ```
     $ wget https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
     ```

     ```
     $ gpg --import NICE-GPG-KEY
     ```

4. Download the packaged components.

- For Rocky8 (x86_64)

  ```
  $ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-el8-
  x86_64.tgz
  ```

- For Rocky8 (ARM aarch64)

  ```
  $ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-el8-
  aarch64.tgz
  ```

- For Amazon Linux 2023, RHEL9, CentOS9, Rocky9 (x86_64)

  ```
  $ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-el9-
  x86_64.tgz
  ```

- For Amazon Linux 2023, RHEL9, CentOS9, Rocky9 (ARM aarch64)

  ```
  $ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-el9-
  aarch64.tgz
  ```

- For Ubuntu22 (x86_64)

  ```
  $ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-ubuntu2204-
  x86_64.tgz
  ```

- For Ubuntu22 (ARM aarch64)

  ```
  $ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-ubuntu2204-
  aarch64.tgz
  ```

- For Ubuntu24 (x86_64)

  ```
  $ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-ubuntu2404-
  x86_64.tgz
  ```

- For Ubuntu24 (ARM aarch64)

  ```
  $ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-ubuntu2404-
  aarch64.tgz
  ```

5. Unzip the packaged components.

```
$ tar -xf nice-dcv-access-console-*.tgz
```

6. Run `ls`, and you should see the following components.

   - **Handler**, **Web Client**, and **Authentication** components – These components end in `.rpm` or `.deb` depending on your distribution.
   - **Setup Wizard Script** – This is a Python script called `wizard.py` to setup the Amazon DCV Access Console.
   - **Setup Wizard Folder** – This folder `access_console_config_wizard` contains the supporting files for the Setup Wizard.
   - **Setup Wizard JSON Files** – These two `.json` files can be used by the Setup Wizard to pre-populate setup parameters. One called `wizard_input.json` and one called `onebox_wizard_input.json`. These can be used by the Setup Wizard to populate setup options.

7. Ensure that the Setup Wizard is set up properly.

```
$ python3 wizard.py --help
```

Preparing the host

For users to visit the Amazon DCV Access Console, the host that the components are installed on needs to be accessible via port 443. Make sure that your host can accept incoming requests on that port from the IP address(es) your users will be connecting from. See the Networking and connectivity for more details.

Note

If you are using SELinux on the host, you need to enable the `httpd_can_network_connect bool` in order for NGINX to forward requests. To do this, run

```
$ sudo setsebool -P httpd_can_network_connect 1
```

Step 2: Run the Setup Wizard

The Setup Wizard will install the components and dependencies for the Access Console, and configure a single host to run all of the Access Console components. For more information on how to use the Setup Wizard see Using the Setup Wizard.

Running the Setup Wizard in interactive mode

Interactive mode is the default setup mode for the Amazon DCV Access Console. It will guide you through the setup process and validate the installation when done.

1. Navigate to the folder where you extracted the Amazon DCV Access Console components.
2. Run the following command:

   ```
   $ python3 wizard.py --is-onebox
   ```

3. Answer the series of questions that appear. These questions determine how to configure the Access Console.

The Setup Wizard will finish by validating that the installation was successful. It will then print the resolvable DNS of the host you provided. The Amazon DCV Access Console will be accessible at that address and any user present on that host will be able to log in.

Running the Setup Wizard in non-interactive mode

Non-interactive mode is the manual setup mode for the Amazon DCV Access Console. This setup allows more configuration in your setup process. You will need to manually fill in the JSON file. See Modifying setup wizard parameters for more details.

1. Go to the file `onebox_wizard_input.json`. This is the JSON file provided with the Wizard.
2. Do one of the following:

   If the Broker is configured on the same host as you are installing the Access Console components, update the following parameters:

   - `onebox-address`– The resolvable DNS of the host that the components are being installed on.
   - `register-with-broker`– Configure to true.

- `show-cookie-link`– If you want to display a link to a cookie disclaimer sign-in on the page, set this parameter to `true`.
- `cookie-link-target`– Set this to the link you want your users to follow for the cookie disclaimer. If you set `show-cookie-link` to `false`, leave it as is.
- `show-privacy-link`– If you want to display a link to a privacy disclaimer on the sign in page, set this parameter to true `true`.
- `privacy-link-target`– Set this to the link you want your users to follow for the privacy disclaimer. If you set `show-privacy-link` to `false`, leave it as is.
- `mariadb-username`– A username you would like to use with MariaDB (if you choose MariaDB as your datastore).
- `mariadb-password`– A password you would like to use the with MariaDB user (if you choose MariaDB as your datastore).
- `admin-user`– The username of a user to grant administrative privileges for the Access Console.

If the Broker is configured on a different host from where you are installing the Access Console components, update the following parameters:

- `onebox-address`– The resolvable DNS of the host that the components are being installed on.
- `broker-address`– The resolvable DNS of the host that the Broker is running on.
- `broker-client-id`– The Broker Client ID that was registered.
- `broker-client-password`– The Broker Client Password that was registered.
- `show-cookie-link`– If you want to display a link to a cookie disclaimer on the sign in page, set this parameter to `true`.
- `cookie-link-target`– Set this to the link you want your users to follow for the cookie disclaimer. If you set `show-cookie-link` to `false`, leave it as is.
- `show-privacy-link`– If you want to display a link to a privacy disclaimer on the sign in page, set this parameter to true `true`.
- `privacy-link-target`– Set this to the link you want your users to follow for the privacy disclaimer. If you set `show-privacy-link` to `false`, leave it as is.
- `mariadb-username`– A username you would like to use with MariaDB (if you choose MariaDB as your datastore).

- `mariadb-password`– A password you would like to use the with MariaDB user (if you choose MariaDB as your datastore).
- `admin-user`– The username of a user to grant administrative privileges for the Access Console.

Setting up on multiple hosts

This section explains how to install the Amazon DCV Access Console components on a multiple hosts. Before proceeding, you must first ensure you have completed the necessary Prerequisites.

Steps

- Step 1: Prepare your environment
- Step 2: Run the Setup Wizard
- Step 3: Install the components

Step 1: Prepare your environment

The Amazon DCV Access Console has three components Handler, Web Client, and Authentication Server. These components can be installed on multiple hosts. See Amazon DCV Access Console Requirements to ensure your setup meets the requirements.

Preparing the components and the Setup Wizard

1. Connect to the host on which you intend to install the Amazon DCV Access Console components.
2. Create a directory where you will save the installation files.

```
$ mkdir dcv-access-console
```

```
$ cd dcv-access-console
```

3. The Amazon DCV Access Console packages are digitally signed with a secure GPG signature. To allow the package manager to verify the package signature, you must import the Amazon DCV GPG key. To do so, open a terminal window and import the Amazon DCV GPG key by entering:

   - For all Linux distributions except Ubuntu::

```
$ sudo rpm --import https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
```

- For Ubuntu:

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
```

```
$ gpg --import NICE-GPG-KEY
```

4. Download the packaged components.

- For Rocky8 (x86_64)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-el8-
x86_64.tgz
```

- For Rocky8 (ARM aarch64)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-el8-
aarch64.tgz
```

- For Amazon Linux 2023, RHEL9, CentOS9, Rocky9 (x86_64)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-el9-
x86_64.tgz
```

- For Amazon Linux 2023, RHEL9, CentOS9, Rocky9 (ARM aarch64)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-el9-
aarch64.tgz
```

- For Ubuntu22 (x86_64)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-ubuntu2204-
x86_64.tgz
```

- For Ubuntu22 (ARM aarch64)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-ubuntu2204-
aarch64.tgz
```

- For Ubuntu24 (x86_64)

  ```
  $ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-ubuntu2404-
  x86_64.tgz
  ```

- For Ubuntu24 (ARM aarch64)

  ```
  $ wget https://d1uj6qtbmh3dt5.cloudfront.net/nice-dcv-access-console-ubuntu2404-
  aarch64.tgz
  ```

5. Unzip the packaged components.

   ```
   $ tar -xf nice-dcv-access-console-*.tgz
   ```

6. Run `ls`, and you should see the following components.

   - **Handler**, **Web Client**, and **Authentication** components – These components end in `.rpm` or `.deb` depending on your distribution.
   - **Setup Wizard Script** – This is a Python script called `wizard.py` to setup the Amazon DCV Access Console.
   - **Setup Wizard Folder** – This folder `access_console_config_wizard` contains the supporting files for the Setup Wizard.
   - **Setup Wizard JSON Files** – These two `.json` files can be used by the Setup Wizard to pre-populate setup parameters. One called `wizard_input.json` and one called `onebox_wizard_input.json`. These can be used by the Setup Wizard to populate setup options.

7. Ensure that the Setup Wizard is set up properly.

   ```
   $ python3 wizard.py --help
   ```

Preparing the hosts

For users to visit the Amazon DCV Access Console, the hosts that the components are installed on needs to be accessible via port 443. Make sure that your host can accept incoming requests on that port from the IP address(es) your users will be connecting from. See Networking and connectivity for more details.

Since we will be using SSL, each host will require a DNS entry pointing to it, and a certificate for that DNS entry.

Preparing the Handler host

This is the host that will communicate with the Session Manager Broker, and will keep track of the state of the Amazon DCV Access Console.

1. Verify the host is able to accept requests from the users on port 443.
2. Verify the host is able to send requests to the Session Manager Broker on the Broker's `client-to-broker-connector-https-port` (port 8443 by default).
3. Take note of the public DNS.
4. Load your certificate onto the instance and take note of the path to the **certificate file**, **key file**, and **keystore** file.

   If you do not already have a certificate, you can create one. For more information, see Generating a self-signed certificate.

Preparing the Authentication Server host

This is the host that will provide the Access Console login page, and create the authorization token the Web Client and Handler use to validate requests.

1. Verify the host is able to accept requests on port 443 from the addresses that your users will be connecting from. Most likely, this will be any address.
2. Take note of the public DNS.
3. Load your certificate onto the instance and take note of the path to the **certificate file**, **key file**, and **keystore** file.

   If you do not already have a certificate, you can create one. For more information, see Generating a self-signed certificate.

Preparing the Web Client host

This is the host that will serve as web application that admins and users will use to connect to the Amazon DCV Access Console.

The Web Client host should also be able to send requests to the hosts the Handler and the Authentication Server are running on. By default, when running on separate hosts, the Handler and Authentication Server run on port 443, although this can be customized.

1. Verify the host is able to accept requests on port 443 from the addresses that your users will be connecting from. Most likely, this will be any address.
2. Take note of the public DNS.
3. Load your certificate onto the instance and take note of the path to the **certificate file**, **key file**, and **keystore** file.

   If you do not already have a certificate, you can create one. For more information, see Generating a self-signed certificate.

Step 2: Run the Setup Wizard

The Setup Wizard will install the components and dependencies for the Access Console, and configure a single host to run all of the Access Console components. For more information on how to use the Setup Wizard see Using the Setup Wizard.

Running the Setup Wizard in interactive mode

Interactive mode is the default setup mode for the Amazon DCV Access Console. It will prompt you for information about the setup, including the DNS entries for each host, the paths to the certificates, and information about the Broker. The Wizard will generate the configuration files and save them to your specified location.

1. Navigate to the folder where you extracted the Amazon DCV Access Console components.
2. Run the following command:

   ```
   $ python3 wizard.py --not-onebox
   ```
3. Answer the series of questions that appear.

   These questions determine how to configure the Access Console.

The Setup Wizard will finish by validating the installation was successful, then print the resolvable DNS of the host you provided. The Amazon DCV Access Console will be accessible at that address and any user present on that host will be able to login.

Running the Setup Wizard in non-interactive mode

Noninteractive mode is the manual setup mode for the Amazon DCV Access Console. This setup allows more configuration in your setup process. You will need to manually fill in the JSON file. For more information on modifying JSON parameters, see Loading a JSON file.

1. Go to the file `wizard_input.json`. This is the JSON file provided with the Wizard.
2. Modify the following parameters:

   - `handler-address`– The resolvable DNS of the host that the Handler will be installed on.
   - `webclient-address`– The resolvable DNS of the host that the Webclient will be installed on.
   - `auth-server-address`– The resolvable DNS of the host that the Authentication Server will be installed on.
   - `broker-address`– The resolvable DNS of the host that the Broker is running on.
   - `broker-client-id`– The Broker Client ID that was registered.
   - `broker-client-password`– The Broker Client Password that was registered.
   - `show-cookie-link`– If you want to display a link to a cookie disclaimer on the sign-in page, set this parameter to `true`.
   - `cookie-link-target`– Set this to the link you want your users to follow for the cookie disclaimer. If you set `show-cookie-link` to `false`, leave it as is.
   - `show-privacy-link`– If you want to display a link to a privacy disclaimer on the sign in page, set this parameter to true `true`.
   - `privacy-link-target`– Set this to the link you want your users to follow for the privacy disclaimer. If you set `show-privacy-link` to `false`, leave it as is.
   - `handler-keystore-password`– The password used by the keystore on the Handler host. Leave it as `changeit` unless you have changed it.
   - `handler-keystore-path`– The path to the keystore file on the Handler host.
   - `auth-server-keystore-password`– The password used by the keystore on the Authentication Server host. Leave it as `changeit` unless you have changed it.
   - `auth-server-keystore-path`– The path to the keystore file on the Handler host.
   - `webclient-cert-path`– The path to the certificate on the Webclient host.
   - `webclient-cert-key-path`– The path to the certificate key on the Webclient host.

- `pam-service-name`– The name of the service to use for PAM authentication on the Authentication Server host. If you are installing on a RedHat-based host, use `system-auth`. If you are using Ubuntu/Debian, use `common-auth`.
- `mariadb-username`– The username of the MariaDB user you created in Step 1 (if you choose MariaDB as your datastore).
- `mariadb-password`– The password you chose for the MariaDB user you created in Step 1 (if you choose MariaDB as your datastore).

Step 3: Install the components

After preparing the Handler, Web Client, and Authentication Server components, you must install them on the hosts you prepared.

Installing the Handler

RHEL, CentOS, Amazon Linux

1. Connect to the host you set up for the Handler.
2. Move the Handler `.rpm` you downloaded to the host in *Step 1: Prepare your environment*.
3. Move the `access-console-handler.properties` and `access-console-handler-secrets.properties` files created by the Setup Wizard to the host.
4. Install the Handler component.

   ```
   $ sudo yum install -y nice-dcv-access-console-handler*.rpm
   ```

5. Move the two `.properties` files to `/etc/dcv-access-console-handler/` and overwrite the existing files.

   ```
   $ sudo mv -f access-console-handler.properties /etc/dcv-access-console-handler/
   access-console-handler.properties
   ```

   ```
   $ sudo mv -f access-console-handler-secrets.properties /etc/dcv-access-console-
   handler/access-console-handler-secrets.properties
   ```

6. Do one of the following:

- If you chose to use DynamoDB as the database, make sure that the instance has permission to access DynamoDB via the Credential Provider Chain, and then skip to the last step.
- If you chose to use MariaDB, you must prepare the database by continuing to the next step.

7. Install MariaDB by doing one of the following:

   - For Amazon Linux 2023

     ```
     $ sudo yum install -y mariadb105-server
     ```

   - For RHEL and CentOS

     ```
     $ sudo yum install -y mariadb-server
     ```

8. Start and enable MariaDB.

   ```
   $ sudo systemctl start mariadb
   ```

   ```
   $ sudo systemctl enable mariadb
   ```

9. Set the **username**, **password**, and **database name** from the previous step.

   ```
   MARIADB_USERNAME=replace with username
   MARIADB_PASSWORD=replace with password
   DATABASE_NAME=replace with database name
   ```

10. Create a new MariaDB user.

    ```
    $ sudo mysql -e "CREATE USER '$MARIADB_USERNAME'@'localhost' IDENTIFIED BY
     '${MARIADB_PASSWORD}'"
    ```

11. Create a new MariaDB database.

    ```
    $ sudo mysql -e "CREATE DATABASE $DATABASE_NAME;"
    ```

12. Grant the user full privileges on the database.

    ```
    $ sudo mysql -e "GRANT ALL PRIVILEGES ON $DATABASE_NAME.* TO
     '$MARIADB_USERNAME'@'localhost';"
    ```

13. Start and enable the Handler component.

```
$ sudo systemctl start nice-dcv-access-console-handler
```

```
$ sudo systemctl enable nice-dcv-access-console-handler
```

Ubuntu, Debian

1. Connect to the host you set up for the Handler.
2. Move the Handler `.deb` file you downloaded to the host in *Step 1: Prepare your environment*.
3. Move the `session-manager-handler.properties` and `session-manager-handler-secrets.properties` files created by the Setup Wizard to the host.
4. Install the Handler component.

   ```
   $ sudo apt install -y nice-dcv-access-console-handler*.deb
   ```

5. Move the two `.properties` files to `/etc/nice-dcv-access-console-handler/` and overwrite the existing files.

   ```
   $ sudo mv -f access-console-handler.properties /etc/dcv-access-console-handler/
   access-console-handler.properties
   ```

   ```
   $ sudo mv -f access-console-handler-secrets.properties /etc/dcv-access-console-
   handler/access-console-handler-secrets.properties
   ```

6. Do one of the following:

   - If you chose to use DynamoDB as the database, make sure that the instance has permission to access DynamoDB via the Credential Provider Chain, and then skip to the last step.
   - If you chose to use MariaDB, you must prepare the database by continuing to the next step.

7. Install MariaDB.

   ```
   $ sudo apt install -y mariadb-server
   ```

8. Start and enable MariaDB.

   ```
   $ sudo systemctl start mariadb
   ```

```
$ sudo systemctl enable mariadb
```

9. Set the **username**, **password**, and **database name** from the previous step.

   ```
   MARIADB_USERNAME=replace with username
   MARIADB_PASSWORD=replace with password
   DATABASE_NAME=replace with database name
   ```

10. Create a new MariaDB user.

    ```
    $ sudo mysql -e "CREATE USER '$MARIADB_USERNAME'@'localhost' IDENTIFIED BY
     '${MARIADB_PASSWORD}'"
    ```

11. Create a new MariaDB database.

    ```
    $ sudo mysql -e "CREATE DATABASE $DATABASE_NAME;"
    ```

12. Grant the user full privileges on the database.

    ```
    $ sudo mysql -e "GRANT ALL PRIVILEGES ON $DATABASE_NAME.* TO
     '$USERNAME'@'localhost';"
    ```

13. Start and enable the Handler component.

    ```
    $ sudo systemctl start dcv-access-console-handler
    ```

    ```
    $ sudo systemctl enable dcv-access-console-handler
    ```

Installing the Authentication Server

RHEL, CentOS, Amazon Linux

1. Connect to the host you set up for the Authentication Server.
2. Move the Authentication Server `.rpm` you downloaded in Step 1: Prepare your environment.
3. Move the `session-manager-auth-server.properties` and `session-manager-auth-server-secrets.properties` files created by the Setup Wizard to the host.
4. Install the Authentication Server component.

```
$ sudo yum install -y nice-dcv-access-console-auth-server*.rpm
```

5. Move the two `.properties` files to `/etc/dcv-access-console-auth-server/` and overwrite the existing files.

```
$ sudo mv -f access-console-auth-server.properties /etc/dcv-access-console-auth-
server/access-console-auth-server.properties
```

```
$ sudo mv -f access-console-auth-server-secrets.properties /etc/dcv-access-console-
auth-server/access-console-auth-server-secrets.properties
```

6. Start and enable the Authentication Server.

```
$ sudo systemctl start dcv-access-console-auth-server
```

```
$ sudo systemctl enable dcv-access-console-auth-server
```

Ubuntu, Debian

1. Connect to the host you set up for the Authentication Server.
2. Move the Authentication Server `.deb` you downloaded to the host in *Step 1: Prepare your environment*.
3. Move the `access-console-auth-server.properties` and `access-console-auth-server-secrets.properties` files created by the Setup Wizard to the host.
4. Install the Authentication Server component.

```
$ sudo apt install -y nice-dcv-access-console-auth-server*.deb
```

5. Move the two `.properties` files to `/etc/dcv-access-console-auth-server/` and overwrite the existing files.

```
$ sudo mv -f access-console-auth-server.properties /etc/dcv-access-console-auth-
server/access-console-auth-server.properties
```

```
$ sudo mv -f access-console-auth-server-secrets.properties /etc/dcv-access-console-
auth-server/access-console-auth-server-secrets.properties
```

6. Start and enable the Authentication Server.

```
$ sudo systemctl start dcv-access-console-auth-server
```

```
$ sudo systemctl enable dcv-access-console-auth-server
```

Installing the Web Client

RHEL, CentOS, Amazon Linux

1. Connect to the host you set up for the Web Client.
2. Move the Web Client `.rpm` you downloaded to the host in *Step 1: Prepare your environment*.
3. Move the `access-console-webclient.properties` and `access-console-webclient-secrets.properties` files created by the Setup Wizard to the host.
4. Move the `dcv-access-console.conf` file created by the Setup Wizard to the host.
5. Install the Web Client component.

```
$ sudo yum install -y nice-dcv-access-console-webclient*.rpm
```

6. Move the two `.properties` files to `/etc/dcv-access-console-webclient/` and overwrite the existing files.

```
$ sudo mv -f access-console-webclient.properties /etc/dcv-access-console-webclient/
access-console-webclient.properties
```

```
$ sudo mv -f access-console-webclient-secrets.properties /etc/dcv-access-console-
webclient/access-console-weblcient-secrets.properties
```

7. Install NGINX.

```
$ sudo yum install -y nginx
```

8. Move the `dcv-access-console.conf` file to `/etc/nginx/conf.d/dcv-access-console.conf`.

   ```
   $ sudo mv dcv-access-console.conf /etc/nginx/conf.d/dcv-access-console.conf
   ```

9. Change the permissions to match the default NGINX configuration file.

   ```
   $ sudo chmod --reference=/etc/nginx/nginx.conf /etc/nginx/conf.d/dcv-access-console.conf
   ```

   ```
   $ sudo chown --reference=/etc/nginx/nginx.conf /etc/nginx/conf.d/dcv-access-console.conf
   ```

10. If you are using SELinux, change the SELinux context to match the default NGINX configuration file.

    ```
    $ sudo chcon --reference=/etc/nginx/nginx.conf /etc/nginx/conf.d/dcv-access-console.conf
    ```

11. Start and enable the Web Client.

    ```
    $ sudo systemctl start dcv-access-console-ui-webclient
    ```

    ```
    $ sudo systemctl enable dcv-access-console-ui-webclient
    ```

12. Start and enable NGINX.

    ```
    $ sudo systemctl start nginx
    ```

    ```
    $ sudo systemctl enable nginx
    ```

Note

If you are using SELinux on the host, you need to enable the `httpd_can_network_connect bool` in order for NGINX to forward requests. To do this, run:

```
$ sudo setsebool -P httpd_can_network_connect 1
```

Ubuntu, Debian

1. Connect to the host you set up for the Web Client.
2. Move the Web Client `.deb` you downloaded to the host in *Step 1: Prepare your environment*.
3. Move the `access-console-webclient.properties` and `access-console-webclient-secrets.properties` files created by the Setup Wizard to the host.
4. Move the `dcv-access-console.conf` file created by the Setup Wizard to the host.
5. Install the Web Client component.

   ```
   $ sudo apt install -y nice-dcv-access-console-webclient*.deb
   ```

6. Move the two `.properties` files to `/etc/dcv-access-console-webclient/` and overwrite the existing files.

   ```
   $ sudo mv -f access-console-webclient.properties /etc/dcv-access-console-webclient/
   access-console-webclient.properties
   ```

   ```
   $ sudo mv -f access-console-webclient-secrets.properties /etc/dcv-access-console-
   webclient/access-console-weblcient-secrets.properties
   ```

7. Install NGINX.

   ```
   $ sudo apt install -y nginx
   ```

8. Move the `dcv-access-console.conf` file to `/etc/nginx/conf.d/dcv-access-console.conf`.

   ```
   $ sudo mv dcv-access-console.conf /etc/nginx/conf.d/dcv-access-console.conf
   ```

9. Start and enable the Web Client.

   ```
   $ sudo systemctl start dcv-access-console-webclient
   ```

```
$ sudo systemctl enable dcv-access-console-webclient
```

10. Start and enable NGINX.

    ```
    $ sudo systemctl start nginx
    ```

    ```
    $ sudo systemctl enable nginx
    ```

Verifying the setup

At this point, the Amazon DCV Access Console should be accessible at the public DNS of the Web Client host. Navigate to `https://`*`web client DNS`* in your web browser. It should redirect to the DNS of the Authentication Server.

If you chose to use PAM authentication, you should be able to log in using the credentials of any user on the host the Authentication Server is running on.

If you chose to use Header-Based Authentication, you will need to modify your request headers using an extension like **Requestly**. You should add a new header with the name being what you chose with the Setup Wizard, and the value as the username you want to log in as.

If you have issues, refer to Troubleshooting.

Generating a self-signed certificate

Every host that is running a Amazon DCV Access Console component needs to have a certificate. If you are bringing your own certificate, you don't need to follow these instructions.

> **Note**
>
> Note that this requires the OpenJDK version 1.8 to be installed on the system.

1. Connect to the host that requires a self-signed certificate.
2. Create a directory to store the certificate.

    ```
    $ sudo mkdir -p /usr/local/var/dcv-access-console/security/
    ```

```
$ cd/usr/local/var/dcv-access-console/security/
```

3. Create the subject of the certificate using the public DNS for the host.

```
$ CERT_SUBJ="/CN=public DNS"
```

4. Set the keystore password. If you have not changed it, the password is `changeit`.

```
$ CERT_PASSWORD="changeit"
```

5. Create the RootCA and use it to sign the certificate.

```
$ sudo openssl req -new -x509 -nodes -newkey rsa:2048 -out rootCA.pem -keyout
 rootCA.key -subj "$CERT_SUBJ" -days 1825
```

```
$ sudo openssl req -new -sha256 -nodes -newkey rsa:2048 -out server.csr -keyout
 server.key -passout pass:$CERT_PASSWORD -subj "$CERT_SUBJ"
```

```
$ sudo openssl x509 -req -sha256 -in server.csr -CA rootCA.pem -CAkey rootCA.key -
CAcreateserial -out server.pem -days 1825
```

6. Create the PKCS12 file.

```
$ sudo openssl pkcs12 -export -nodes -in server.pem -inkey server.key -
out keystore.p12 -name server -passin pass:$CERT_PASSWORD -password pass:
$CERT_PASSWORD
```

7. Import the RootCA and the certificate into the keystore.

```
$ sudo keytool -import -alias rootca -cacerts -storepass $CERT_PASSWORD -file
 rootCA.pem -noprompt
```

```
$ sudo keytool -import -alias server -cacerts -storepass $CERT_PASSWORD -file
 server.pem -noprompt
```

Take note of the paths to:

- `server.pem`

- `server.key`
- `keystore.p12`
- `rootCA.pem`

You will need them during configuration.

Getting started with the Amazon DCV Session Manager console

The following topic describes how to use the Session Manager console.

Topics

- Accessing the console

Accessing the console

After successfully setting up the console, you can access it from a custom URL, configured during setup, from a web browser. See the Requirements for a list of supported web browsers.

Levels of access

There are two levels of access that you might have when using the console.

- **Owner** – You created the session. You may be an admin or a user.
- **Administrator** – You are the admin, and are viewing a session created by one of your users. You have the same permissions as the owner, including connecting to and closing the session.

Logging in to the Session Manager Console

From the Console home page, you can log in using your Amazon DCV credentials.

If you have trouble logging in, do one of the following:

- **If you are an administrator** – You must debug the Auth Server. For more information, see Log in Errors in Troubleshooting.
- **If you are a user** – Contact your administrator for assistance.

Using the Amazon DCV Access Console

The following topic describes how to use the Amazon DCV Access Console.

Topics

Sessions

A session is a span of time when the Amazon DCV server is able to accept connections from a client. Each session has a specified owner and set of permissions.

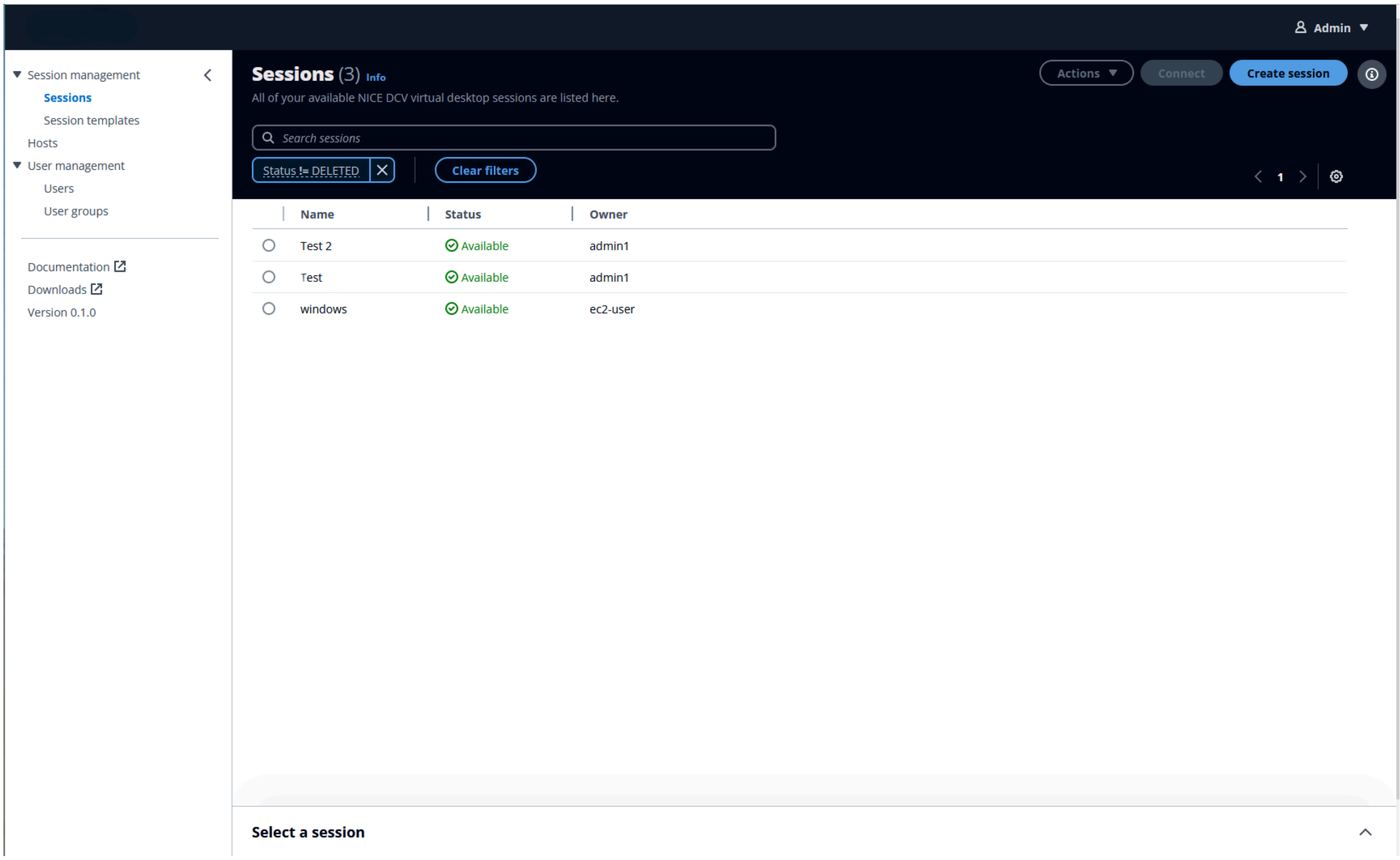
Before your clients can connect to a Amazon DCV session, you must create a Amazon DCV session on the Amazon DCV server. When you create a Amazon DCV session, you change the state of the server to accept connections from a client. Amazon DCV supports both console and virtual sessions.

On the **Sessions** page, you can view sessions that you created yourself, and the detailed session information. If there are no available sessions, you must choose **Create session** to begin.

> **Note**
>
> If you experience issues accessing the sessions created outside of the console, you may need to debug that or delete that session.

You can configure the visible fields in the top navigation bar by selecting the gear icon. To view more details in a split panel view, use the picker to select a session. Then select the caret (^) icon at the bottom-right corner of the page.

By default, sessions that have been closed are hidden with a filter. You can remove the filter to see previously closed sessions.



Sessions (3)

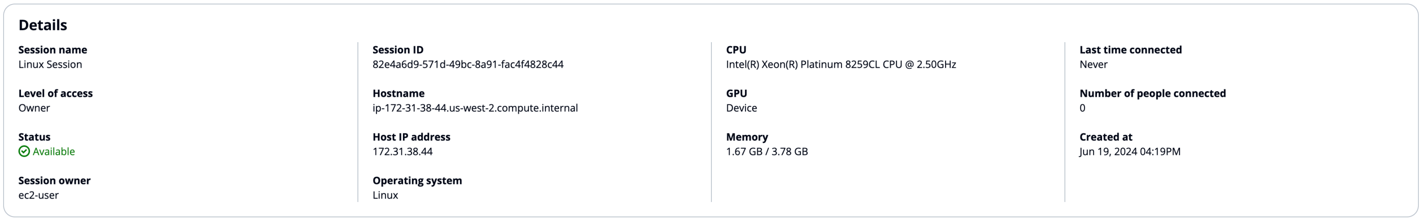
All of your available NICE DCV virtual desktop sessions are listed here.

	Name	Status	Owner
○	Test 2	✓ Available	admin1
○	Test	✓ Available	admin1
○	windows	✓ Available	ec2-user

Select a session

Session details

This includes the session parameters themselves. For more information, see DescribeSessions. The details also include the Amazon DCV server information that the session is placed on. For more information, see DescribeServers.



Details

Session name
Linux Session

Level of access
Owner

Status
✓ Available

Session owner
ec2-user

Session ID
82e4a6d9-571d-49bc-8a91-fac4f4828c44

Hostname
ip-172-31-38-44.us-west-2.compute.internal

Host IP address
172.31.38.44

Operating system
Linux

CPU
Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz

GPU
Device

Memory
1.67 GB / 3.78 GB

Last time connected
Never

Number of people connected
0

Created at
Jun 19, 2024 04:19PM

Property	Description
Session name	The session name. This field can't be changed after creation (Name in the DescribeSessions API).

Property	Description
Level of access	The level of access for a particular session.
Status	A session has four states associated with it. • **Creating –** The Broker is in the process of creating the session. • **Available –** The session is ready to accept client connections (maps to "READY" in the DescribeSessions API). • **Closing –** The session is being closed (maps to "DELETING" in the DescribeSessions API). • **Closed –** The session is closed (maps to "DELETED" in the DescribeSessions API). • **Unknown –** Unable to determine the session's state. The Broker and the Agent might be unable to communicate. Contact your administrator for help troublesh ooting.
Session owner	The name of the session owner (`Owner` is in the DescribeSessions API).
Session ID	The unique ID of the session (`Id` is in the DescribeSessions API).

Property	Description
Hostname	The hostname of the host server that the Amazon DCV server is running on (`Servers.Hostname` in the DescribeServers API).
Host IP address	The unique IP of the Amazon DCV server (`Servers.ID` in the DescribeServers API).
Operating systems	The name of the host server operating system that the Amazon DCV server is running on (`Host.OS.Family` in the DescribeServers API).
CPU	Information about the host server's CPU that the Amazon DCV server is running on (`Host.CpuInfo.ModelName` in the DescribeServers API).
GPU	Information about the host server's GPU that the Amazon DCV server is running on (`ModelName` in the DescribeServers API).
Memory	Information about the host server's memory, in gigabytes. This information is displayed as [Used GB/Total GB] (`Memory.UsedBytes/Memory/TotalBytes` in the DescribeServers API).
Last time connected	The last time a user connected to this session (`LastDisconnectionTime` in the DescribeSessions API).

Property	Description
Number of people connected	The number of people currently connected to this session (`NumOfConnections` in the DescribeSessions API).
Created at	The time that the session was created at (`CreationTime` in the DescribeSessions API).

Topics

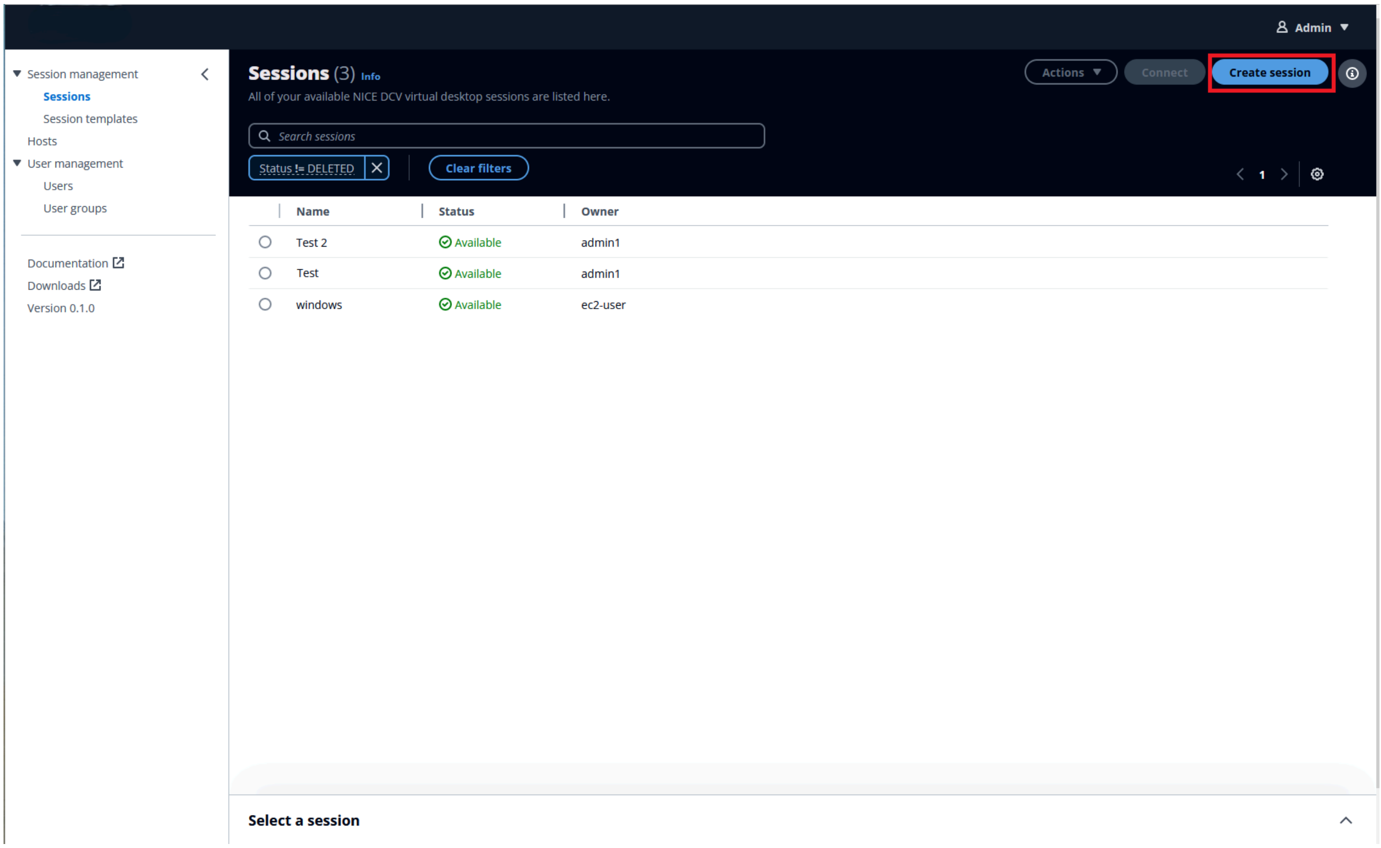
- Creating a session
- Connecting to a session
- Closing a session

Creating a session

To use this console, you must create a session. A session is a span of time when the Amazon DCV server can accept connections from a client. By creating a new session, your default level of access is **owner**, which gives you admin permissions.

To create a new session, you must select a template already provided by the administrator. Session templates are specified parameters that you can create a session with. If there are no templates available to choose from, contact the administrator to create a template and assign it to you.

1. Select **Sessions** under the **Session management** tab.
2. Select the **Create session** button.



Sessions (3) Info

All of your available NICE DCV virtual desktop sessions are listed here.

Status != DELETED

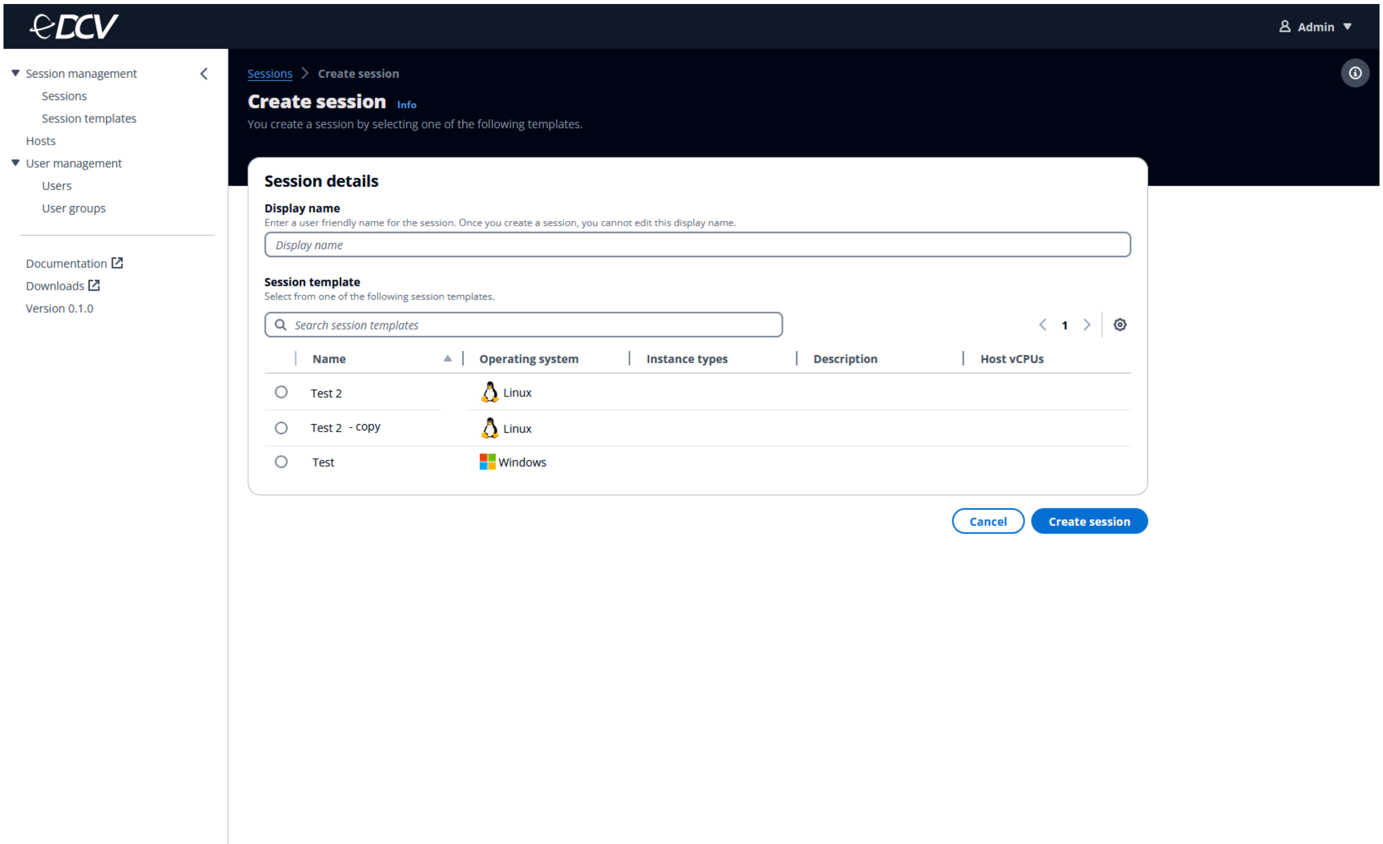
	Name	Status	Owner
○	Test 2	Available	admin1
○	Test	Available	admin1
○	windows	Available	ec2-user

3. In **Display name**, enter a user friendly name for your session.

> **Note**
>
> After you create a session, you can't edit this name.

4. Select a **Session template**.

5. Select the **Create session** button.



Create session

You create a session by selecting one of the following templates.

Session details

Display name

Enter a user friendly name for the session. Once you create a session, you cannot edit this display name.

Display name

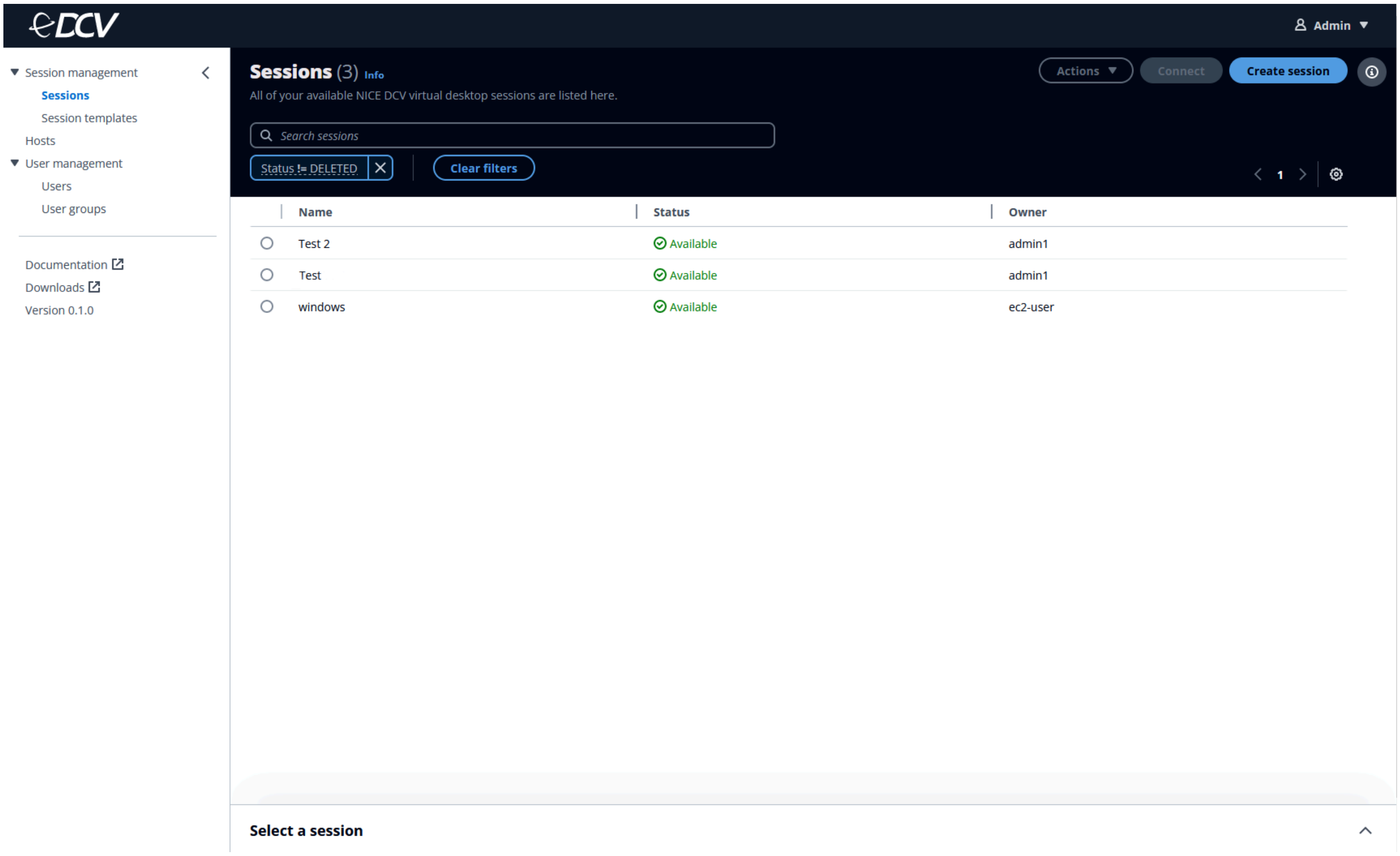
Session template

Select from one of the following session templates.

Search session templates

	Name	Operating system	Instance types	Description	Host vCPUs
○	Test 2	Linux			
○	Test 2 - copy	Linux			
○	Test	Windows			

The newly created session will appear in the Sessions dashboard. It may take a few minutes to create the session. In that time, you won't be able to connect to or close the session.



Sessions (3) Info

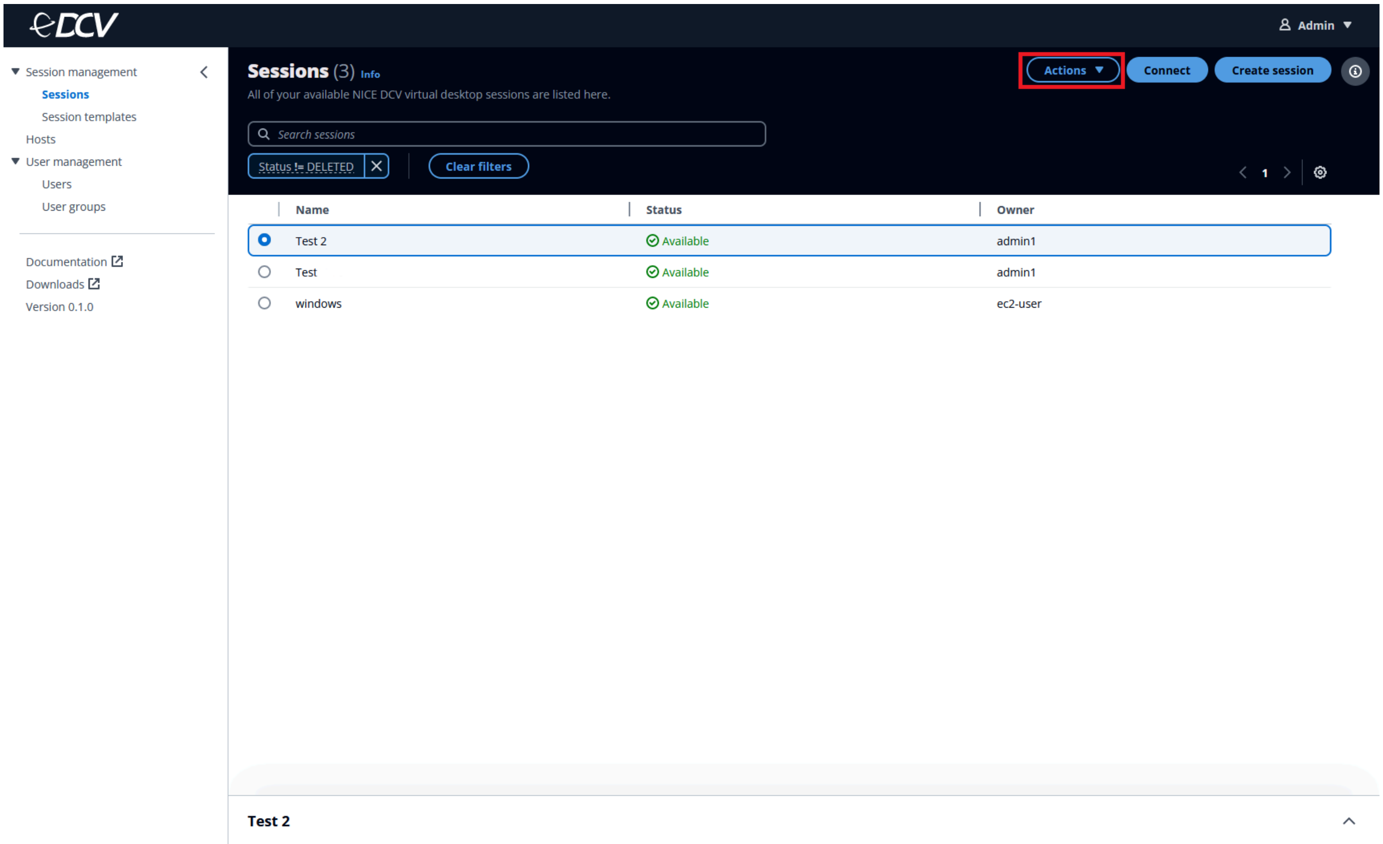
All of your available NICE DCV virtual desktop sessions are listed here.

	Name	Status	Owner
○	Test 2	✓ Available	admin1
○	Test	✓ Available	admin1
○	windows	✓ Available	ec2-user

Connecting to a session

You can connect to a session after it has been created. You can connect to a session from either the DCV web client, or a native Windows or macOS client application.

1. Select the **Actions** button in the session window that you want to view.



All of your available NICE DCV virtual desktop sessions are listed here.

	Name	Status	Owner
●	Test 2	Available	admin1
○	Test	Available	admin1
○	windows	Available	ec2-user

2. Select **Connect using** from the menu.

3. Choose from one of the following options:

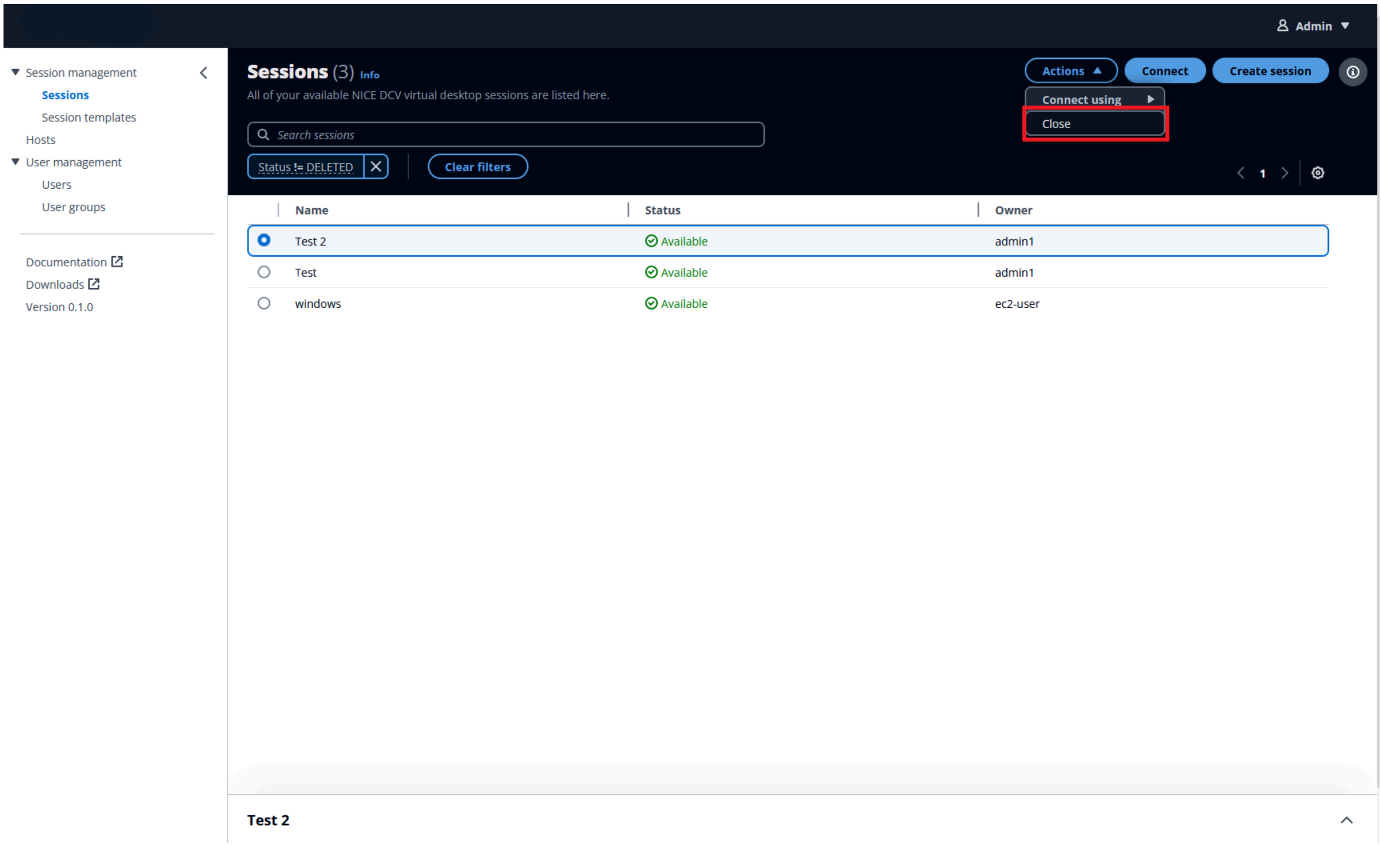
   - **Web browser**— Connects to your session using a web browser.
   - **Windows client**— Connects to your session using the Windows client with the Amazon DCV app. If you don't have the appropriate local Amazon DCV Viewer application downloaded, you will be directed to the Amazon DCV download site where you can download the latest viewer.
   - **macOS client**— Connects to your session using the macOS client with the Amazon DCV app. If you don't have the appropriate local Amazon DCV Viewer application downloaded, you will be directed to the Amazon DCV download site where you can download the latest viewer.

Closing a session

After you’re completely done with your work, you can **Close** a session and release the underlying resource back to the host server.

> **Note**
>
> Closing a session can't be undone. All locally saved work will be lost. Closing a session doesn't shut down the underlying host server.

1. Go to the **Sessions** page.
2. Select the session that you want to close.
3. Click the **Actions** button in the session window.
4. Select **Close** from the menu.



All of your available NICE DCV virtual desktop sessions are listed here.

	Name	Status	Owner
●	Test 2	Available	admin1
○	Test	Available	admin1
○	windows	Available	ec2-user

5. Select **Close** from the window that appears.

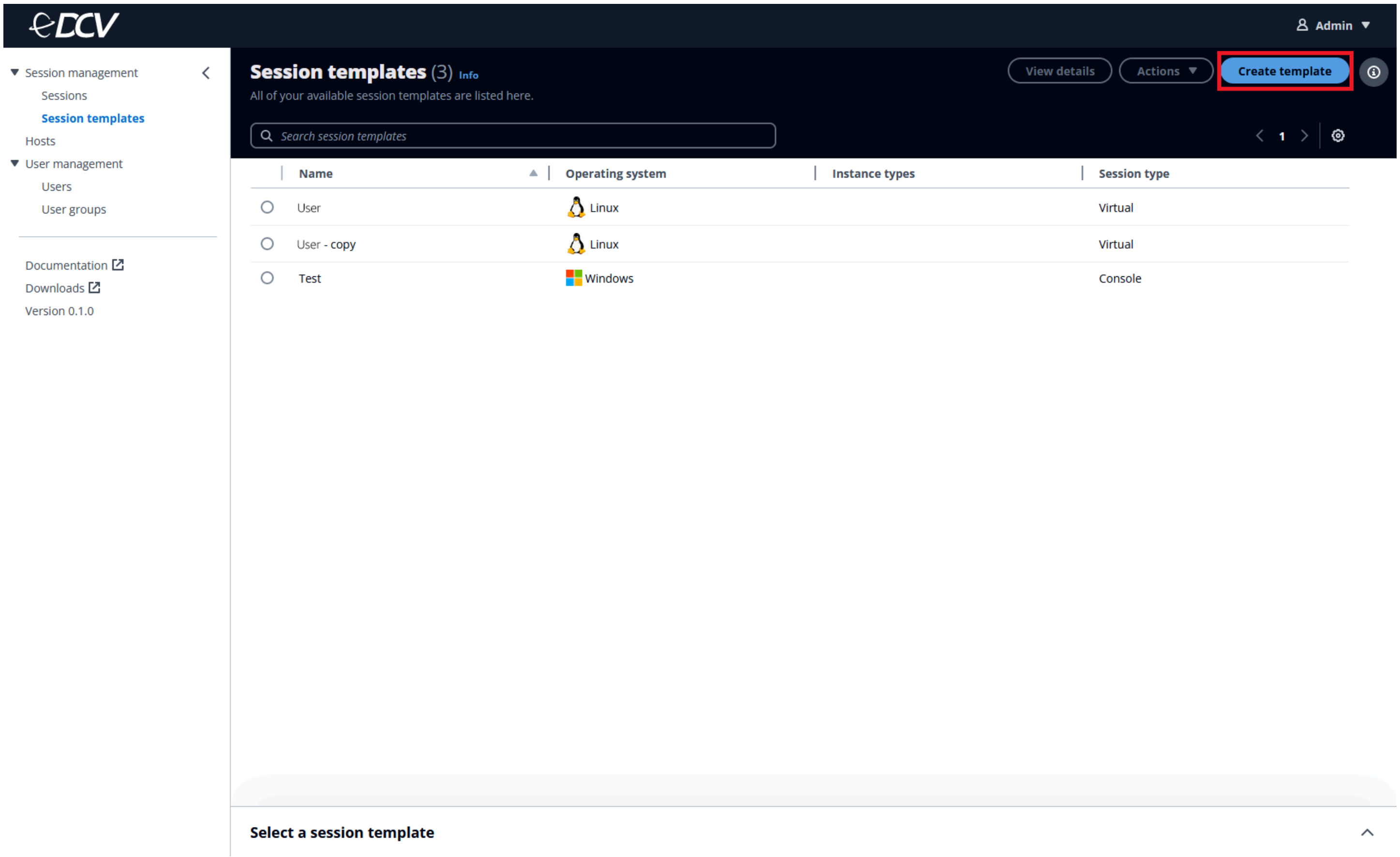
Session templates

A Amazon DCV session template is created by admins to define the details of the session to be created.

To create a session, you must first have an existing session template that you will use to create a session from.

On the **Session templates** page, you can view session templates that you created, and their detailed information.

You can configure the visible fields in the top navigation bar by selecting the gear icon. To view more details in a split panel view, use the picker to select a template, and then select the caret (^) icon at the bottom-right corner of the page.



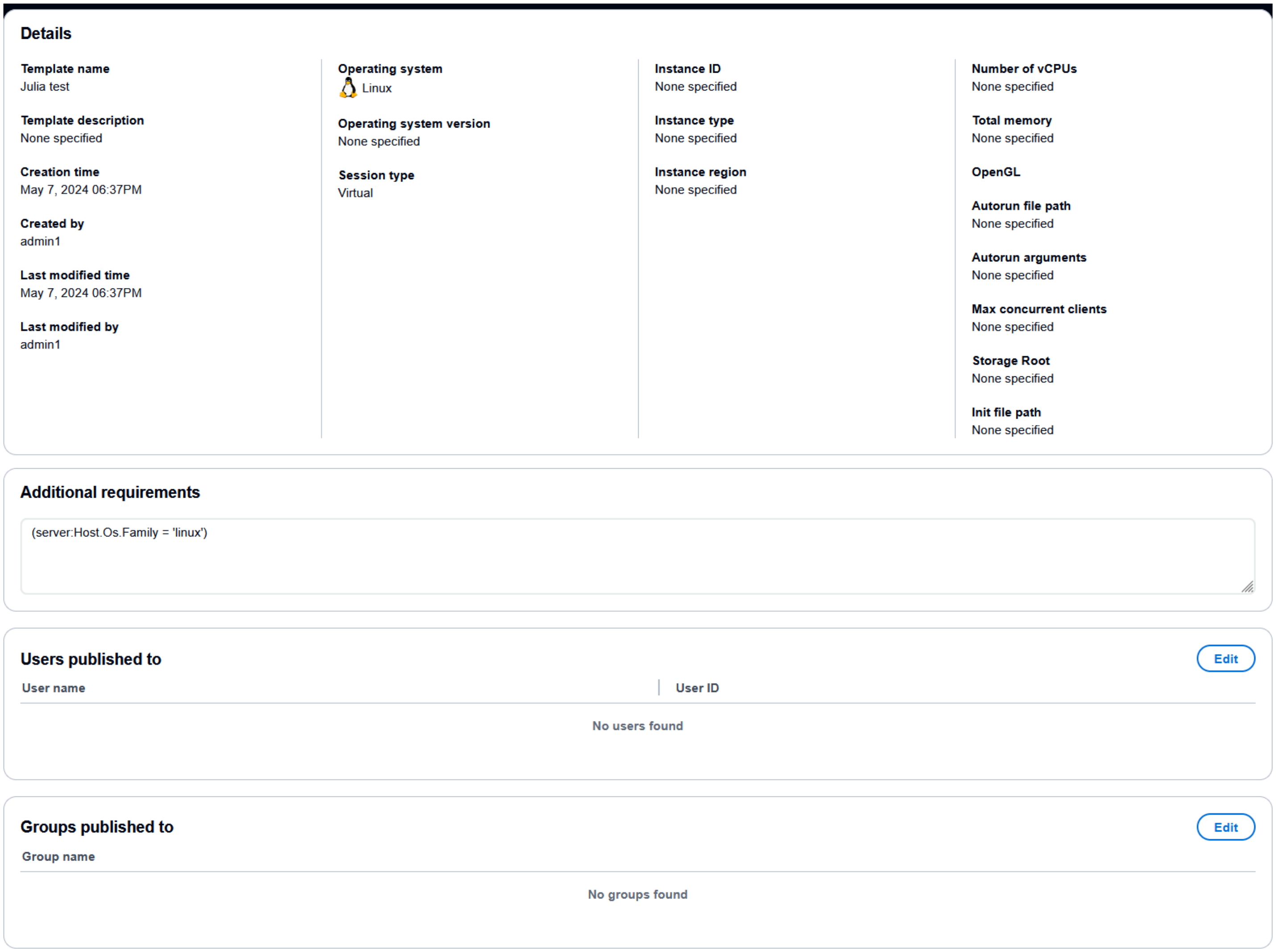
Session templates (3) Info

All of your available session templates are listed here.

	Name	Operating system	Instance types	Session type
○	User	Linux		Virtual
○	User - copy	Linux		Virtual
○	Test	Windows		Console

Session template details

For more information see Creating a session.



Details

Template name Julia test	Operating system Linux	Instance ID None specified	Number of vCPUs None specified
Template description None specified	Operating system version None specified	Instance type None specified	Total memory None specified
Creation time May 7, 2024 06:37PM	Session type Virtual	Instance region None specified	OpenGL
Created by admin1			Autorun file path None specified
Last modified time May 7, 2024 06:37PM			Autorun arguments None specified
Last modified by admin1			Max concurrent clients None specified
			Storage Root None specified
			Init file path None specified

Additional requirements

(server:Host.Os.Family = 'linux')

Users published to

User name	User ID
No users found	

Groups published to

Group name
No groups found

Property	Description
Template name (required)	The descriptive name that's shown to users.
Template description	The session template description. This is to describe the intended use case for the template, and help users choose the appro priate template.
Operation system (required)	The operating system family of the host server that the Amazon DCV server runs on. This

Property	Description
	must be either a Linux, Windows or macOS operating system (`Host.OS.Family` in the `DescribeServers` API).
Operating system version	The version of the operating system of the host server that the Amazon DCV server is running on (`Host.Os.Version` in the `DescribeServers` API).
Session type (required)	The session type, which must either be a console or a virtual session. A **console session** is supported on macOS, Linux and Windows servers, and will be the only session on the specified server. A **virtual session** is supported only on Linux servers, and allows multiple sessions on the specified server (Type in the `CreateSessions` `API`). For more information about the types of sessions, see Introduction to Amazon DCV Sessions in the *Amazon DCV Administrator Guide*.
OpenGL (Linux virtual only)	Indicates whether the virtual session is configured to use the hardware-based OpenGL. OpenGL stands for Open Graphics Library, and is a set of standard APIs used to interface with graphics processing hardware, allowing hardware acceleration through the GPU. OpenGL is supported with virtual sessions only. This parameter isn't supported with Windows Amazon DCV servers (`DcvGlEnabled` in the `CreateSessions` API).

Property	Description
Instance ID	The ID of the Amazon EC2 instance. This parameter only applies for customers hosting on AWS, and will not be shown to customers hosting on-premises (`Host.Aws.Ec2InstanceId` in the `DescribeServers` API).
Instance type	The type of Amazon EC2 instance. This parameter only applies for customers hosting on AWS, and will not be shown to customers hosting on-premise (`Host.Aws.Ec2InstanceType` in the `DescribeServers` API).
Instance Region	The AWS Region of the Amazon EC2 host. This parameter only applies for customers hosting on AWS, and will not be shown to customers hosting on-premises (`Host.Aws.Region` in the `DescribeServers` API).
Host vCPU	The number of virtual CPUs on the host server (`Host.CpuInfo.NumberOfCpus` in the `DescribeServers` API).
Host memory in bytes	The total memory, in bytes, on the host server that the Amazon DCV server is running on (`Host.Memory.TotalBytes` in the `DescribeServers` API).

Property	Description
Autorun file path (Windows and Linux virtual only)	The path to a file on the host server that runs inside the session. The file path is relative to the autorun directory specified for the `agent.autorun_folder` Agent configuration parameter. If the file is in the specified autorun directory, specify the file name only. If the file isn't in the specified autorun directory, specify the relative path. For more information, see Agent configuration file in the *Amazon DCV Session Manager Administrator Guide*. The file is run on behalf of the specified **owner**. The specified owner must have permission to run the file on the server. On Windows Amazon DCV servers, the file is run when the owner logs in to the session. On Linux Amazon DCV servers, the file is run when the session is created. Console sessions on Windows Amazon DCV servers and virtual sessions on Linux Amazon DCV servers are supported. Console sessions on macOS Amazon DCV servers and Linux Amazon DCV servers are not supported. (`AutorunFile` in the `CreateSessions` API).

Property	Description
Autorun arguments (Linux virtual only)	Command line arguments passed to **AutorunFile** upon its execution inside the session. Arguments are passed in the order that they appear into the given array. Maximum allowed number of arguments and maximum allowed length of each argument can be configured. For more information, see Broker configuration file in the *Amazon DCV Session Manager Administrator Guide*. Virtual sessions on Linux Amazon DCV servers are supported. Console sessions on macOS, Windows and Linux Amazon DCV servers are not supported (`AutorunFileArguments` in the `CreateSessions` API).
Max concurrent clients	The maximum number of concurrent Amazon DCV clients allowed to connect to the session at a given time. To specify that there is no maximum, enter 0. (`AutorunFileArguments` in the `CreateSessions` API).

Property	Description
Init file path (Linux virtual only)	The path to a folder on the host server used to store custom scripts allowed to initialize Amazon DCV server sessions when they are created. The file path is relative to the init directory specified for the `agent.init_folder` Agent configuration parameter. If the file is in the specified init directory, specify the file name only. If the file isn't in the specified init directory, specify the relative path. The folder must be accessible and the files must be executable by users who use the **InitFile** request parameter of the **CreateSessions** API. For more information, see [Create Sessions]() in the *Amazon DCV Session Manager Developer Guide* or [Agent configuration]() file in the *Amazon DCV Session Manager Administrator Guide*. Virtual sessions on Linux Amazon DCV servers are suppoprted. Console sessions on macOS, Windows and Linux Amazon DCV servers are not supported (`InitFile` in the `CreateSessions` API).

Property	Description
Storage root	Specifies the path to the folder used for session storage. Session storage is a folder on the Amazon DCV server that clients can access when they're connected to a specific Amazon DCV session. When you enable session storage for a session, clients can download files from, and upload files to, the specified folder. This feature enables clients to share files while connected to a session. For more information, see Create Sessions in the *Amazon DCV Session Manager Developer Guide* or Enabling Session Storage in the *Amazon DCV Administrator Guide* (`StorageRoot` in the `CreateSessions` API).
Additional host server requirements	Use this text box to set the requirements that the server must satisfy to place the session. The requirements can include server tags and/or server properties, both server tags and server properties are retrieved by calling the **DescribeServers** API. Requirements support both condition and Boolean expressions.

Some of these settings you've already specified elsewhere in the **Configure** step (like Operating System). Those settings are pre-populated in the additional requirements box, and are immutable from the text box itself. To change those settings, you must change them from the specified UI elements. You can also add and modify additional requirements using the syntax provided in the Create Session documentation. For a complete list of supported server properties, see Create Session in the *Amazon DCV Session Manager Developer Guide*.

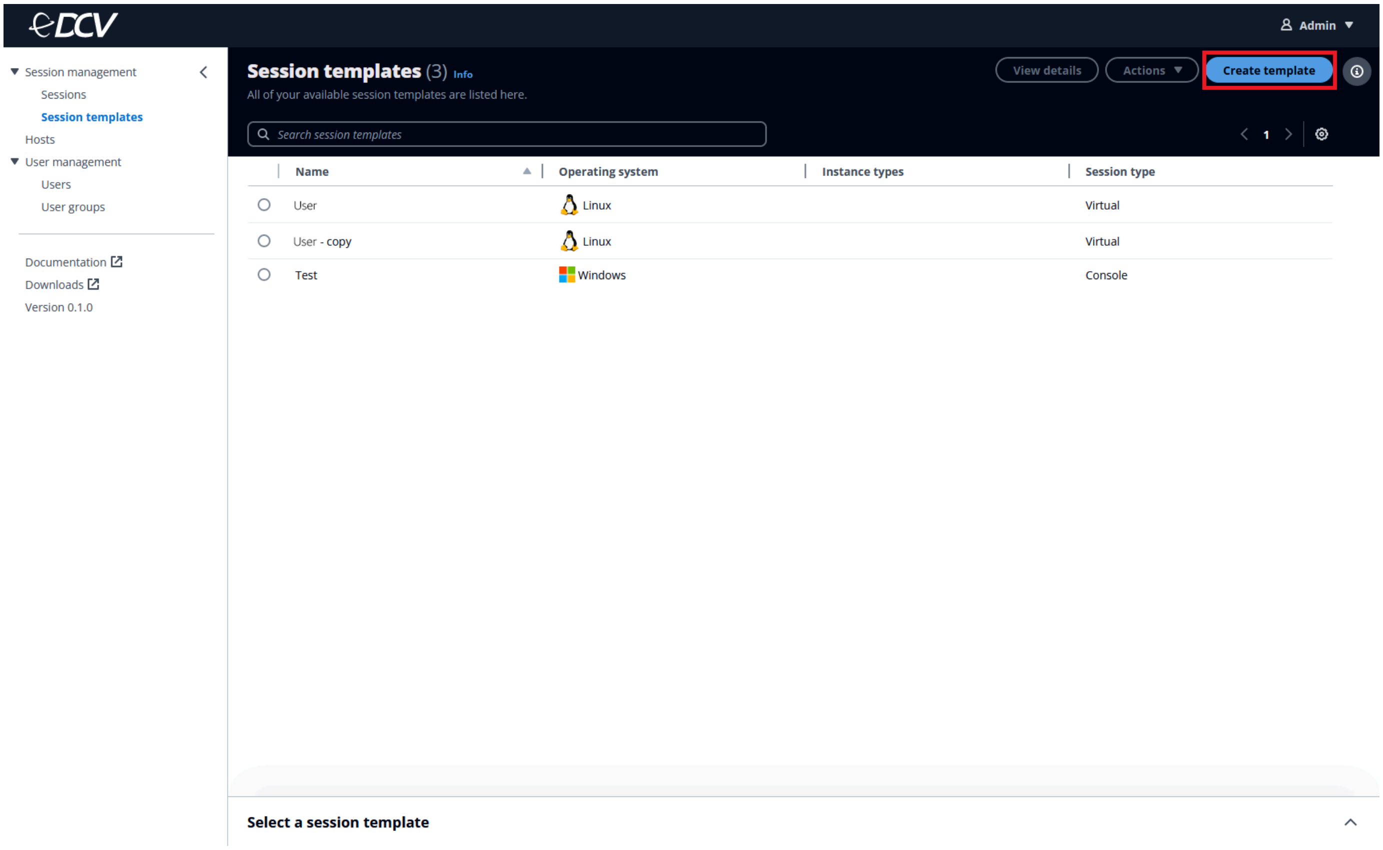
Topics

- Duplicating a session template
- Editing a session template
- Deleting a session template

Creating a session template

A session template is required to create sessions within the console. The session template sets the parameters and details of the session.

1. Go to the **Session templates** page.
2. Select the **Create template** button.



Session templates (3) Info

All of your available session templates are listed here.

	Name	Operating system	Instance types	Session type
○	User	Linux		Virtual
○	User - copy	Linux		Virtual
○	Test	Windows		Console

Select a session template

3. Enter the information in the **Configure template details** page.

   This page chooses the parameters of your session template. These parameters define the details of the session and create boundaries for what kind of hosts a session can be created on. See Session template details for more information.

4. Assign users or user groups to the session template.

You can assign a session template for existing users or groups when creating sessions. You can do this either during template creation or after a template has already been created. For more information, see Assigning a session template to users or groups.

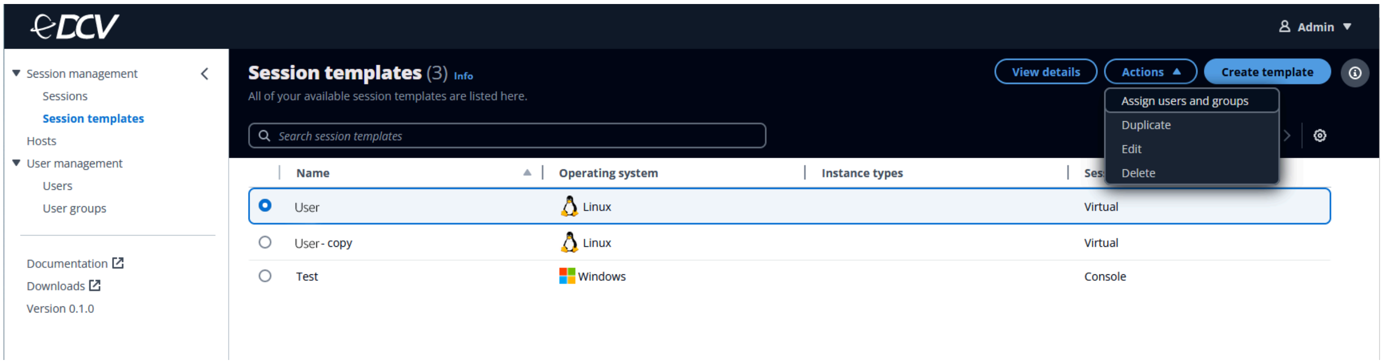
5. Select the **Next** button.
6. Review the template details for accuracy. To change the template, select **Edit** to go back to the **Configure template details** page.
7. Select the **Create template** button.

Assigning a session template to users or groups

In order for users to create sessions, they must first have a session template assigned to them.

You may assign a session template to users or groups either during the original template creation process or after a template has already been created. See Creating a session template.

1. Select the session template that you want to assign.
2. Click on the **Actions** button.
3. Select **Assign users and groups** from the menu.

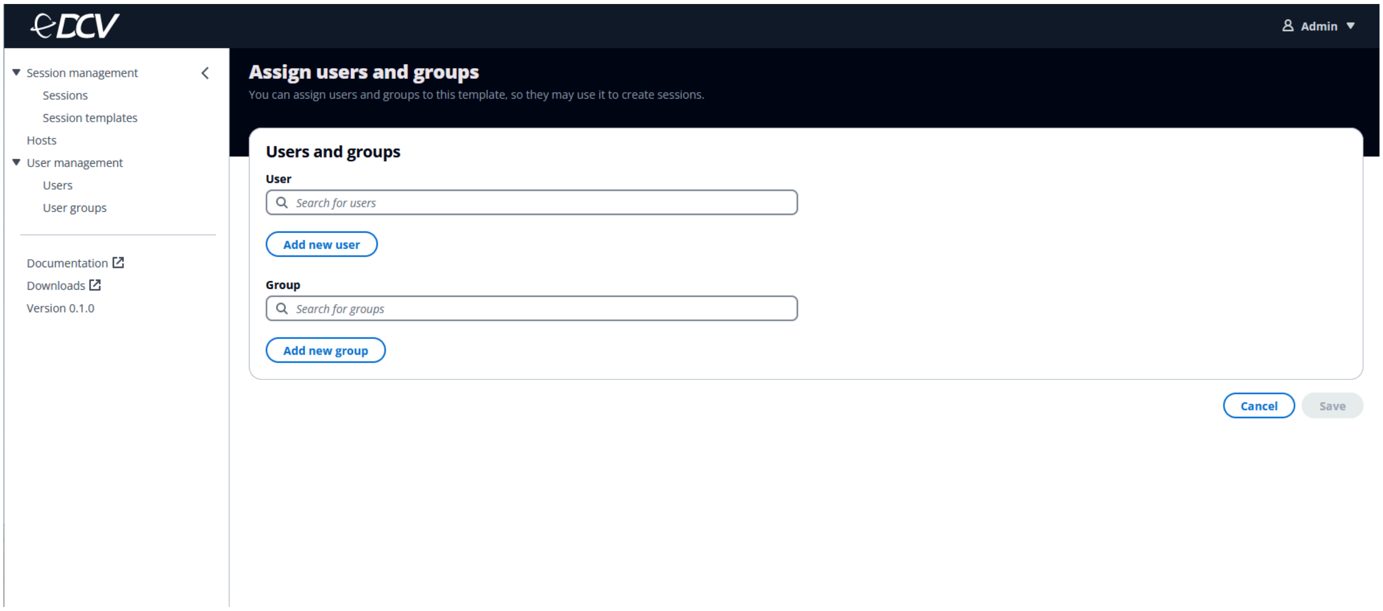


Session templates (3) Info

All of your available session templates are listed here.

	Name	Operating system	Instance types	Ses
●	User	Linux		Virtual
○	User - copy	Linux		Virtual
○	Test	Windows		Console

4. Enter the name of the user in the **User** field or the name of the user group in the **Group** field.



Assign users and groups

You can assign users and groups to this template, so they may use it to create sessions.

Users and groups

User

Search for users

Group

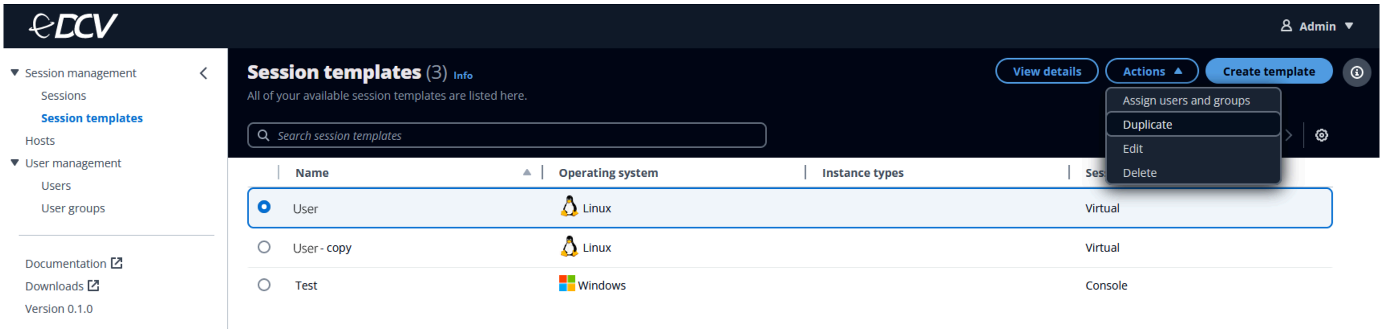
Search for groups

5. Click on either the **Add new user** or **Add new group** button.
6. Choose **Save**.

Duplicating a session template

Instead of creating a new session template, you can choose to duplicate an existing session template and change its parameters to your specifications.

1. Select the session template that you want to duplicate.
2. Click on the **Actions** button.
3. Select **Duplicate** from the drop-down menu. This will take you to the **Configure template details** page.



Session templates (3) Info

All of your available session templates are listed here.

	Name	Operating system	Instance types	Ses
●	User	Linux		Virtual
○	User - copy	Linux		Virtual
○	Test	Windows		Console

4. Change any of the information in the **Configure template details** page.

This page chooses the parameters of your session template. These parameters define the details of the session and create boundaries for what kind of hosts a session can be created on. See Session template details for more information.

5. Assign users or user groups to the session template.

   You can assign a session template for existing users or groups to use when creating sessions. You can do this either during template creation or after a template has already been created. For more information, see Assigning a session template to users or groups.

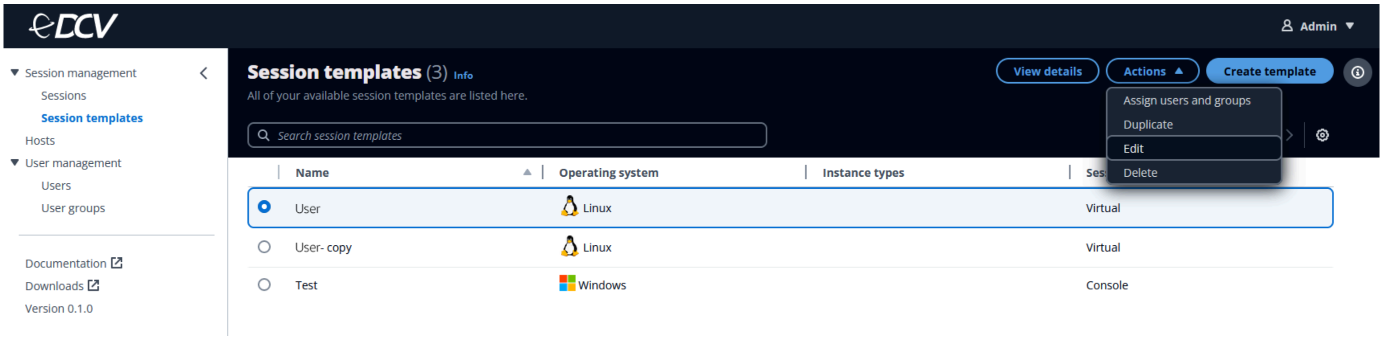
6. Select the **Create template** button.

Editing a session template

If you need to adjust any sessions details, you can edit the parameters of an existing session template.

> **Note**
>
> Editing an existing template could affect users already assigned to it. Any changes you make will not affect the sessions already created. However, it will affect users the next time they create a session using the modified template. If you do not want to affect users who already have this template assigned to them, Duplicating a session template may be a better option.

1. Select the session template that you want to edit.
2. Click on the **Actions** button.
3. Select **Edit** from the drop-down menu. This will take you to the **Configure template details** page.



	Name	Operating system	Instance types	Ses
●	User	Linux		Virtual
○	User- copy	Linux		Virtual
○	Test	Windows		Console

4. Change any of the information in the **Configure template details** page.

   This page chooses the parameters of your session template. These parameters define the details of the session and create boundaries for what kind of hosts a session can be created on. See Session template details for more information.

5. Assign users or user groups to the session template.

   You can assign a session template for existing users or groups to use when creating sessions. You can do this either during template creation or after a template has already been created. For more information, see Assigning a session template to users or groups.

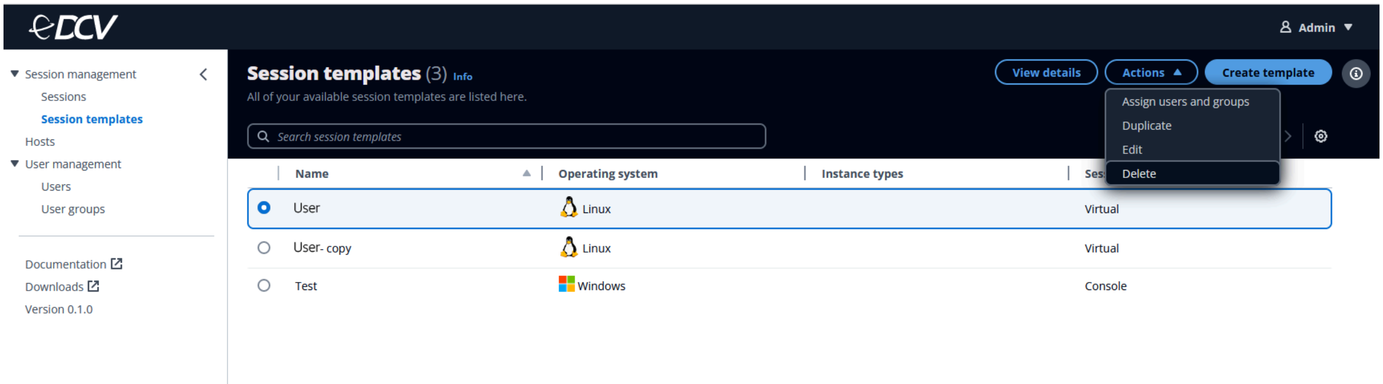
6. Select the **Update template** button.

Deleting a session template

You can delete a session template when you're completely done with it.

> **Note**
>
> Deleting a session can't be undone. Active sessions that were created with a deleted template won't be affected. However, any assigned users will no longer see the template available when they create a new session.

1. Select the session template that you want to delete.
2. Click on the **Actions** button.
3. Select **Delete** from the drop-down menu.



Name	Operating system	Instance types	Ses
User	Linux		Virtual
User- copy	Linux		Virtual
Test	Windows		Console

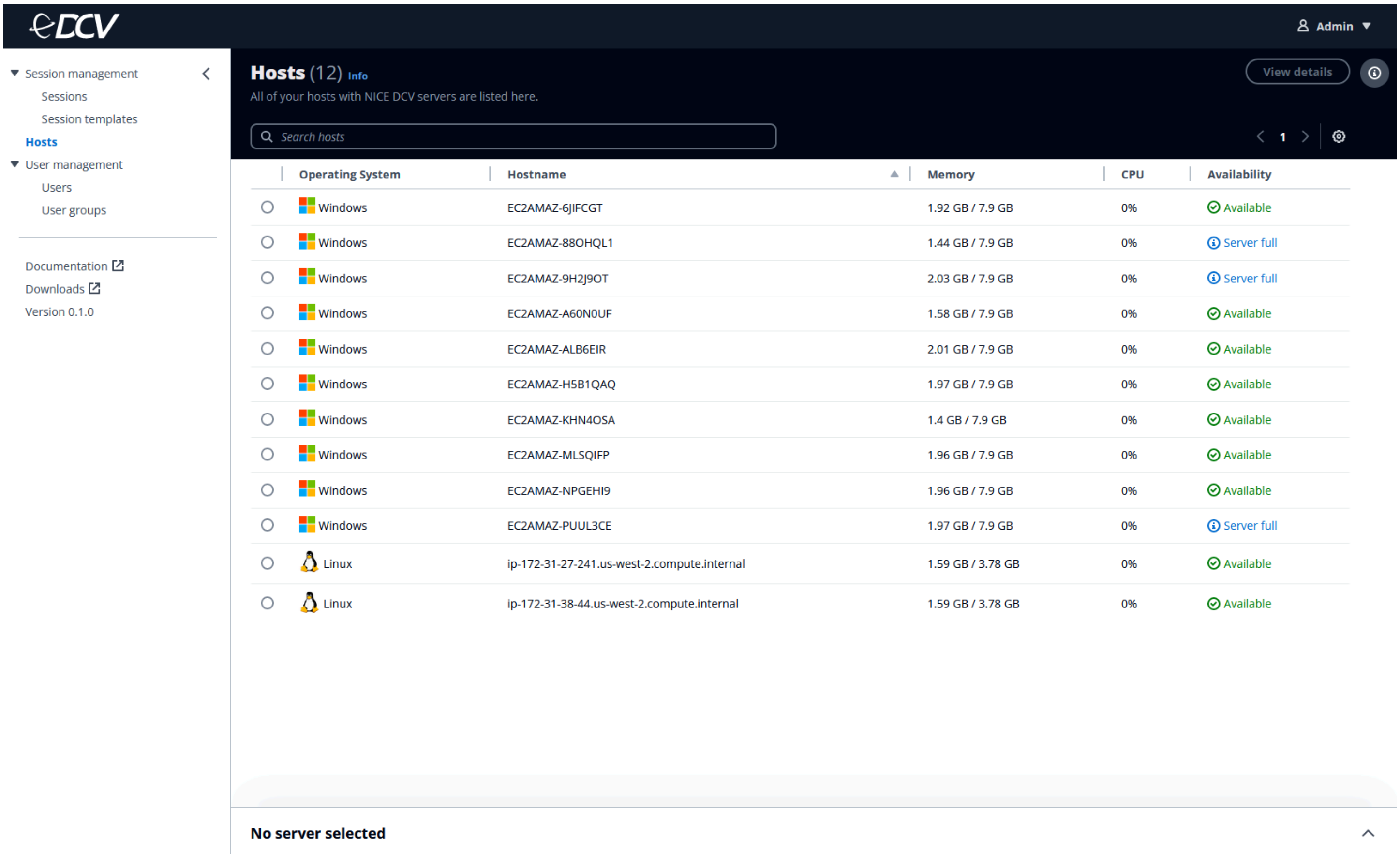
4. Click on the **Delete** button in the window that appears.

Hosts

On the **Hosts** page, you can view a list of host machines (either cloud or on-premises) you have installed Amazon DCV servers configured with Amazon DCV Session Manager.

Before your users can connect to a Amazon DCV session, you must have hosts available for users to create sessions on. You can't spin up hosts, install Amazon DCV servers on hosts, or configure them with the Amazon DCV Session Manager from the console. For more information about installing Amazon DCV servers, see Installing the Amazon DCV server.

You can configure the visible fields in the top navigation bar by selecting the gear icon. To view more details in a split panel view, select a session and then click the caret (^) icon at the bottom-right corner of the page.



Hosts (12) Info

All of your hosts with NICE DCV servers are listed here.

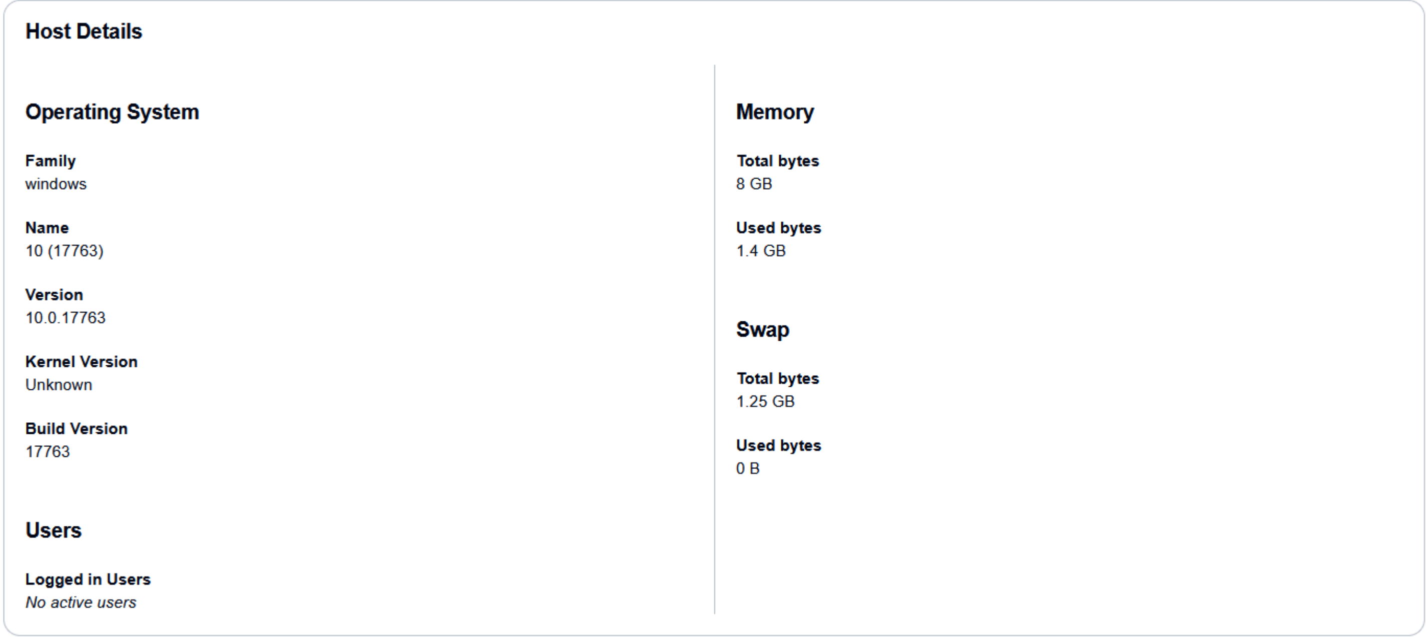
	Operating System	Hostname	Memory	CPU	Availability
○	Windows	EC2AMAZ-6JIFCGT	1.92 GB / 7.9 GB	0%	✓ Available
○	Windows	EC2AMAZ-88OHQL1	1.44 GB / 7.9 GB	0%	ⓘ Server full
○	Windows	EC2AMAZ-9H2J9OT	2.03 GB / 7.9 GB	0%	ⓘ Server full
○	Windows	EC2AMAZ-A60N0UF	1.58 GB / 7.9 GB	0%	✓ Available
○	Windows	EC2AMAZ-ALB6EIR	2.01 GB / 7.9 GB	0%	✓ Available
○	Windows	EC2AMAZ-H5B1QAQ	1.97 GB / 7.9 GB	0%	✓ Available
○	Windows	EC2AMAZ-KHN4OSA	1.4 GB / 7.9 GB	0%	✓ Available
○	Windows	EC2AMAZ-MLSQIFP	1.96 GB / 7.9 GB	0%	✓ Available
○	Windows	EC2AMAZ-NPGEHI9	1.96 GB / 7.9 GB	0%	✓ Available
○	Windows	EC2AMAZ-PUUL3CE	1.97 GB / 7.9 GB	0%	ⓘ Server full
○	Linux	ip-172-31-27-241.us-west-2.compute.internal	1.59 GB / 3.78 GB	0%	✓ Available
○	Linux	ip-172-31-38-44.us-west-2.compute.internal	1.59 GB / 3.78 GB	0%	✓ Available

No server selected

Host information

For more information about the requirements and details of the Amazon DCV servers, see Amazon DCV Servers and DescribeServers.

Host Details



Host Details

Operating System

Family
windows

Name
10 (17763)

Version
10.0.17763

Kernel Version
Unknown

Build Version
17763

Users

Logged in Users
No active users

Memory

Total bytes
8 GB

Used bytes
1.4 GB

Swap

Total bytes
1.25 GB

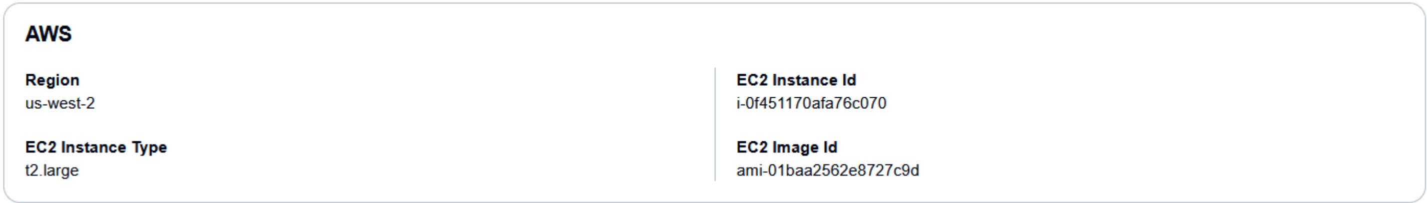
Used bytes
0 B

Property	Description
Family	The host operating system family that the Amazon DCV server is running on, such as Windows, Linux or macOS (`Host.OS.Family` in the DescribeServers API API).
Hostname	The hostname of the host server that the Amazon DCV server is running on (`Servers.Hostname` in the DescribeServers API API).
Name	The name of the host server operating system that the Amazon DCV server is running on (`Host.OS.Name` in the DescribeServers API API).
Version	

Property	Description
	The version of the host server operating system that the Amazon DCV server is running on (`Host.OS.Version` in the DescribeServers API API).
Kernel version	(Linux only) The kernel version of the host server operating system that the Amazon DCV server is running on (`Host.OS.KernelVersion` in the DescribeServers API API).
Build number	(Windows only) The build number of the host server operating system that the Amazon DCV server is running on (`Host.OS.BuildNumber` in the DescribeServers API API).
LoggedInUsers	The usernames of the users that are currently logged into the host server (`Host.OS.LoggedInUsers` in the DescribeServers API API).
Memory	Information about the host server's memory, in gigabytes. This information is displayed as [Used GB/Total GB] (`Memory.UsedBytes / Memory/TotalBytes` in the DescribeServers API).
Memory - Total bytes	The total memory, in bytes, on the host server that the Amazon DCV server is running on (`Memory.TotalBytes` in the DescribeServers API API).

Property	Description
Memory - Used bytes	The used memory, in bytes, on the host server that the Amazon DCV server is running on (`Memory.UsedBytes` in the DescribeServers API API).
Swap - Total bytes	The total swap file size, in bytes, on the host server that the Amazon DCV server is running on (`Swap.TotalBytes` in the DescribeServers API API).
Swap - Used bytes	The used swap file size, in bytes, on the host server that the Amazon DCV server is running on (`Swap.UsedBytes` in the DescribeServers API API).

AWS information



AWS

Region
us-west-2

EC2 Instance Type
t2.large

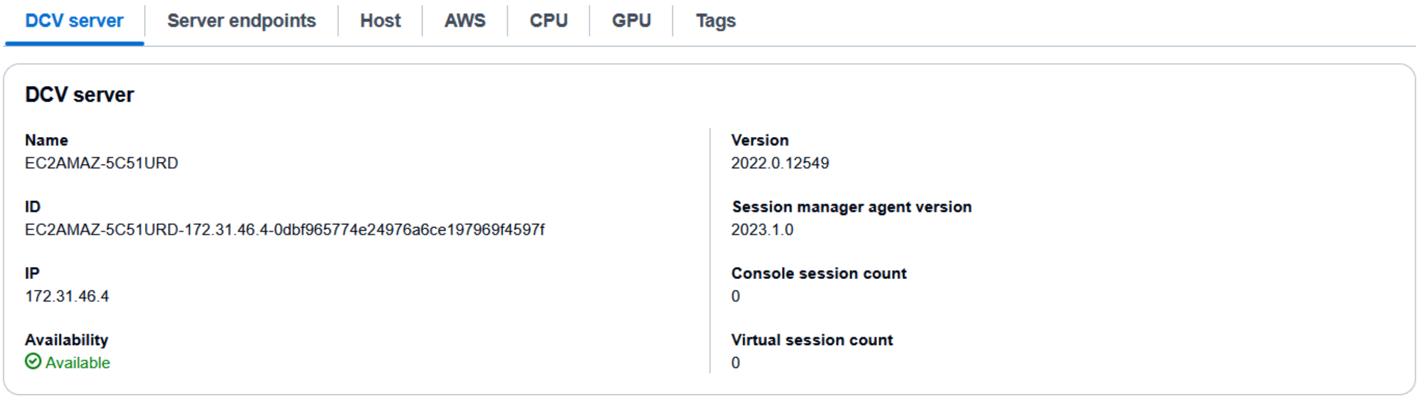
EC2 Instance Id
i-0f451170afa76c070

EC2 Image Id
ami-01baa2562e8727c9d

Property	Description
Region	The Region of the Amazon EC2. This parameter only applies for customers hosting on AWS, and will not be shown to customers hosting on-premise (`Host.Aws.Region` in the `DescribeServers` API).
EC2 Instance Type	

Property	Description
	The type of Amazon EC2 instance. This parameter only applies for customers hosting on AWS, and will not be shown to customers hosting on-premise (`Host.Aws.Ec2InstanceType` in the `DescribeServers` API).
EC2 Image ID	The ID of the Amazon EC2 image. This parameter only applies for customers hosting on AWS, and will not be shown to customers hosting on-premise (`Host.Aws.Ec2IMAGEId` in the `DescribeServers` API).

Amazon DCV server



DCV server

Name
EC2AMAZ-5C51URD

ID
EC2AMAZ-5C51URD-172.31.46.4-0dbf965774e24976a6ce197969f4597f

IP
172.31.46.4

Availability
Available

Version
2022.0.12549

Session manager agent version
2023.1.0

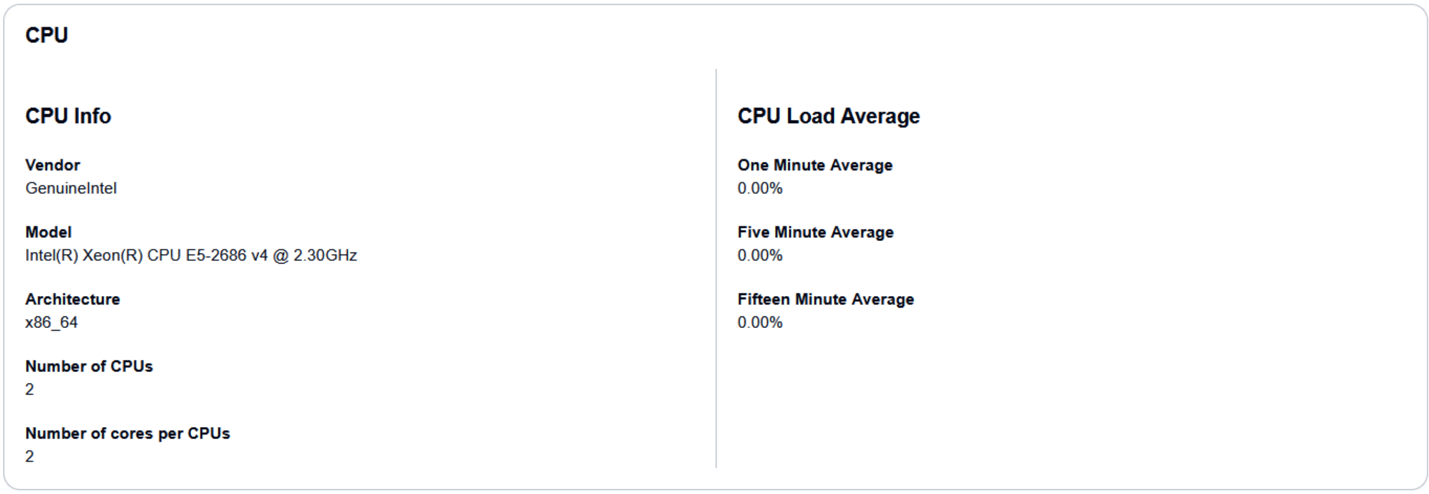
Console session count
0

Virtual session count
0

Property	Description
ID	The unique ID of the Amazon DCV server (`Servers.Id` in the `DescribeServers` API).
Availability	The availability of the Amazon DCV server (`Servers.Availability` in the

Property	Description
	`DescribeServers` API). Possible values include: - AVAILABLE — The server is available and ready for session placement. - UNAVAILABLE — The server is unavailable and can't accept session placement.
Version	The version of the Amazon DCV server (`Servers.Version` in the `DescribeServers` API).
Session Manager agent version	The version Session Manager agent running on the Amazon DCV server (`Servers.SessionManagerAgentVersion` in the `DescribeServers` API).
Console session count	The number of console sessions on the Amazon DCV server (`Servers.ConsoleSessionCount` in the `DescribeServers` API).
Virtual session count	The number of virtual sessions on the Amazon DCV server (`Servers.ConsoleSessionCount` in the `DescribeServers` API).

CPU



CPU

CPU Info

Vendor
GenuineIntel

Model
Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz

Architecture
x86_64

Number of CPUs
2

Number of cores per CPUs
2

CPU Load Average

One Minute Average
0.00%

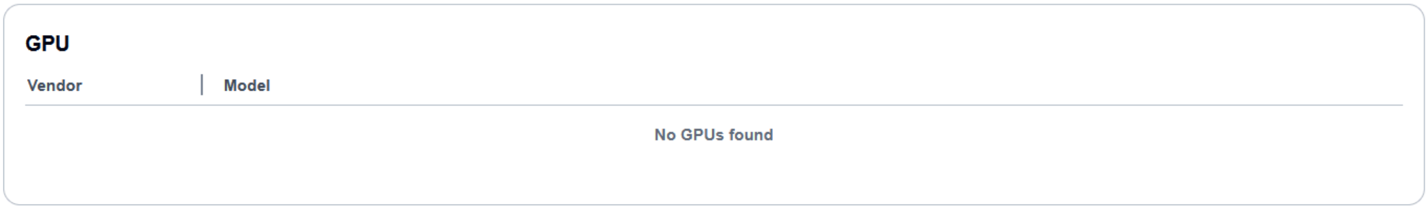
Five Minute Average
0.00%

Fifteen Minute Average
0.00%

Property	Description
Vendor	The vendor of the host server's CPU (`Host.CpuInfo.Vendor` in the `DescribeServers` API).
Model	The model name of the host server's CPU (`Host.CpuInfo.ModelName` in the `DescribeServers` API).
Architecture	The architecture of the host server's CPU (`Host.CpuInfo.Architecture` in the `DescribeServers` API).
Number of vCPUs	The number of virtual CPUs on the host server (`Host.CpuInfo.NumberOfCpus` in the `DescribeServers` API).
Number of physical cores per CPU	The number of physical CPUs on the host server.

Property	Description
One minute average	The average CPU load over the last 1 minute period of the host server (`Host.CpuLoadAverage.OneMinute` in the `DescribeServers` API).
Five minute average	The average CPU load over the last 5 minute period of the host server (`Host.CpuLoadAverage.FiveMinutes` in the `DescribeServers` API).
Fifteen minute average	The average CPU load over the last 15 minute period of the host server (`Host.CpuLoadAverage.FifteenMinutes` in the `DescribeServers` API).

GPU

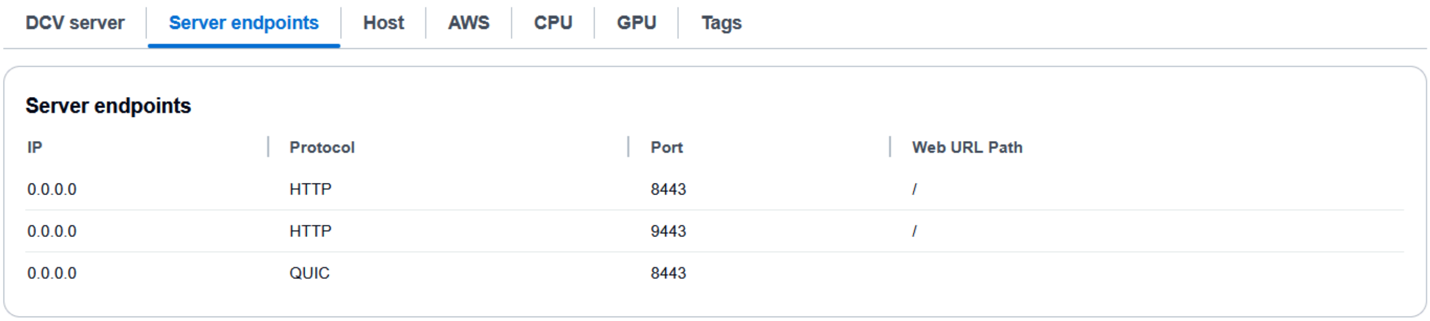

GPU

Vendor	Model

No GPUs found

Property	Description
Vendor	The vendor of the host server's GPU (`Host.Gpus.Vendor` in the `DescribeServers` API).
Model	The model name of the host server's GPU (`Host.Gpus.ModelName` in the `DescribeServers` API).

Server endpoints



Server endpoints

IP	Protocol	Port	Web URL Path
0.0.0.0	HTTP	8443	/
0.0.0.0	HTTP	9443	/
0.0.0.0	QUIC	8443	

Property	Description
IP	The IP address of the Amazon DCV server endpoint (`Servers.Endpoints.IpAddress` in the `DescribeServers` API).
Protocol	The protocol used by the Amazon DCV server endpoint (`Servers.Endpoints.Protocol` in the `DescribeServers` API). Possible values include: • HTTP — The endpoint uses the WebSocket (TCP) protocol. • QUIC — The endpoint uses the QUIC (UDP) protocol.
Port	The port of the Amazon DCV server endpoint (`Servers.Endpoints.Port` in the `DescribeServers` API).
Web URL path	The web URL path of the Amazon DCV server endpoint. Available for the HTTP protocol

Property	Description
	only (`Servers.Endpoints.WebUrlPath` in the `DescribeServers` API).
Tags	The tags assigned to the host server that the Amazon DCV server is running on (`Host.Tags` in the `DescribeServers` API).

Managing users in the Amazon DCV Access Console

The following section explains how to manage users and groups with the Amazon DCV Access Console.

Topics

Importing users and groups

Users will only appear in the Amazon DCV Access Console if they have been directly imported from the Access Console, or have already logged in. Users are imported into the Access Console by uploading a CSV file. Once imported, user names populate on the **Users** page of the Access Console.

User groups can also be imported with a CSV file to the Access Console. If you choose not to import user groups, you can create from the Access Console directly.

To import users and groups with a CSV file

1. Go to the **Users** page.
2. Select the **Import users** button.
3. Upload a CSV file where each row has the following format:

   userId,loginUsername,displayName,role,groupIds

   With the following parameters:

   - **userId**– This field is required.
   - **loginUsername**– This field is optional. It will be set to the same as userId, if left empty. When using external OAuth, this should match the username from the OAuth provider.
   - **displayName**– This field is optional. It will be set to the same as userId, if left empty.
   - **role**– This field is optional, and can be set to either Admin or User. It will be set to User, if left empty.

- **groupIds**– This field is optional. You can include multiple GroupIDs, separated by “**|**”.

> **Note**
>
> You can import users and groups from the same CSV file.

User and group assignment from OAuth claims

Importing users by CSV is not supported for some external OAuth setups because the userId may not be known until first login. When using external OAuth, you can configure the Amazon DCV Access Console to automatically assign roles and groups to users based on claims in the OAuth token.

The following properties in the `access-console-handler.properties` file enable role and group assignment from OAuth claims:

- `jwt-default-groups-claim-key` is the key that contains the default groups. To enable access to session templates on first login, create groups and assign them session templates through the Access Console, then use the group IDs in your external OAuth configuration. Groups are only assigned on first login and subsequent group management should be done through the Access Console. The groups value should be comma-separated group IDs (e.g., "group1,group2"). If a group does not exist, it will be created.
- `jwt-role-claim-key` is the key that contains the user's role. The role value must match a configured role (e.g., Admin, User, Guest). Invalid roles fall back to the default role defined in `default-role` in the Handler configuration file.

For more information about these and other Handler configuration parameters, see Handler configuration files.

Users

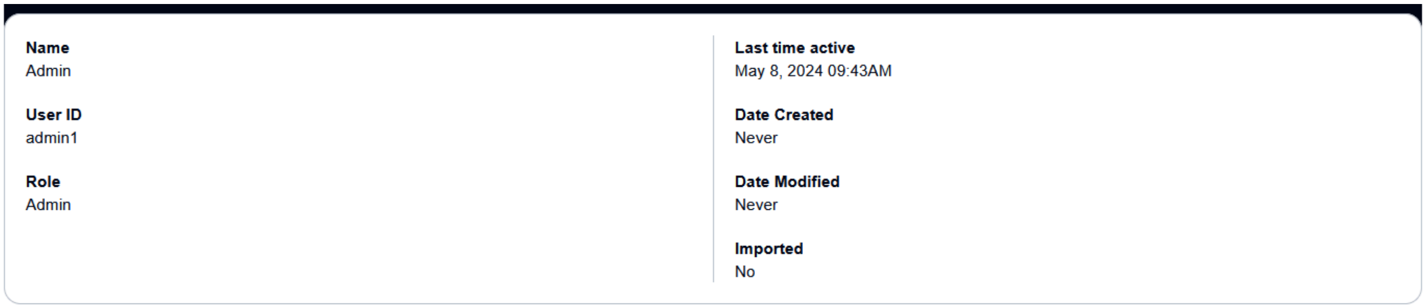
The Amazon DCV Access Console allows admins to manage users, their roles and their access to the Console. You cannot edit a user’s name or any of their parameters or delete a user directly from the Console.

On the **Users** page, you can view the users saved in your datastore and their detailed information. Users appear here if they have been directly imported from the Access Console, or have already logged in. For a complete list of users that are authorized to log into the Access Console, you must refer to your externally configured users datastore. For more information on how to configure your datastore, see Datastore.

Before your users can connect to the Access Console, you must configure either Pluggable Authenticate Modules (PAM) Authentication, or HTTP Header authentication. See Authentication Methods for more information.

User details

On the bottom part of the screen, the details for the selected user is displayed. This graphic shows which details are displayed.



Name
Admin

User ID
admin1

Role
Admin

Last time active
May 8, 2024 09:43AM

Date Created
Never

Date Modified
Never

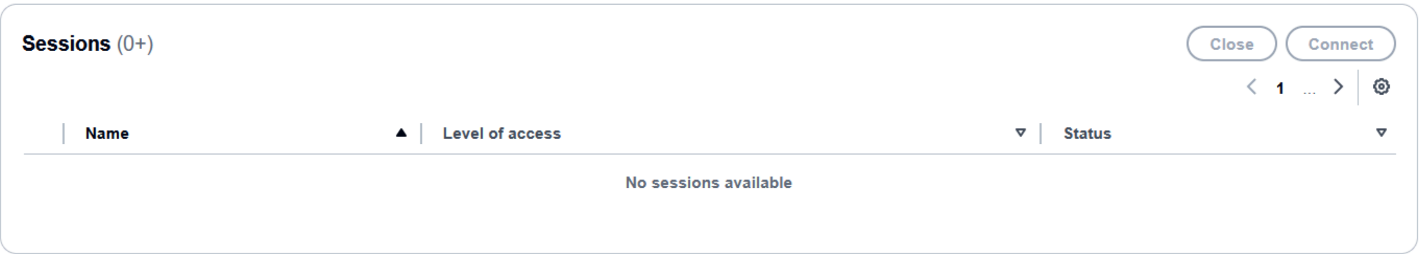
Imported
No

Property	Description
Name	The display name of the user.
User ID	The unique ID of the user.
Role	The role a user can have when using the Access Console - admin or user.
Last time active	The last time the user connected to the Access Console.
Date created	

Property	Description
	The date the user was created in the Access Console.
Date modified	The last date that the user was modified in the Access Console.
Imported	Indicates whether or not the user was manually imported to the Access Console.

Session

These are the active sessions that the user has created. Its parameters are listed below.



Sessions (0+)

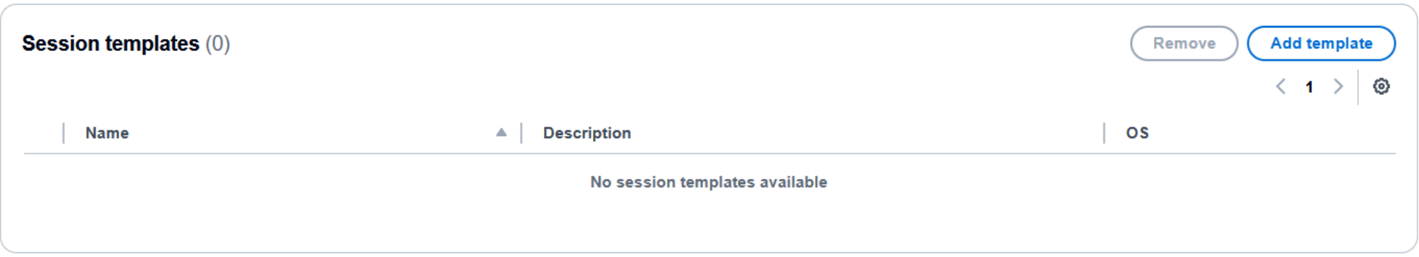
Name	Level of access	Status

No sessions available

Property	Description
Name	The display name of the user.
Level of access	Whether the user is set to Administrator or User.
Status	The current status of the user.

Session template

These are the session templates that are available for the user. Its parameters are listed below.



Session templates (0)

Name	Description	OS

No session templates available

Property	Description
Name	The name of the session template.
Description	The description of the session template.
OS	The operating system of the session template.

User roles

There are two roles a user can have with the Amazon DCV Access Console: admin and user. Both of these user roles can create and connect to their own sessions.

Admin role

- Create sessions
- View and connect to all sessions
- View, create, assign and modify session templates
- View hosts
- View and import users
- View, import and modify user groups

User role

- Create sessions
- View and connect to all sessions

Changing a user's role

To change a user’s role, you must edit the user directly from your configured datastore. You cannot change a user's role from the Access Console.

DynamoDB

1. Navigate to the users table in the DynamoDB console.
2. Select **Explore Table Items**.
3. Select the entry that corresponds to the user you want to be an admin.
4. Select **Actions** then **Edit item**.
5. Modify the role to be **Admin** or **User**.
6. Select **Save and close**.
7. Connect to the Handler host.
8. Restart the Handler.

```
$ sudo systemctl restart dcv-access-console-handler
```

MariaDB

1. Connect to the Handler host.
2. Enter the username of the user you want to be an admin:

```
ADMIN_USER=replace with username
```

3. Enter the database name you chose during setup. If you left it as the default, the name is `dcv_access_console`.

```
DATABASE_NAME=replace with database name
```

4. Retrieve the name of the users table.

```
$ sudo mysql -e "show tables like '%User';" --database=$DATABASE_NAME
```

   It is the table ending in User, not `SessionTemplatePublishedToUser`.

5. Update the user role.

```
USER_TABLE=user table name
```

```
$ sudo mysql -e "UPDATE $DATABASE_NAME.$USER_TABLE SET role = 'Admin' WHERE
 userId='$ADMIN_USER';"
```

6. Restart the Handler.

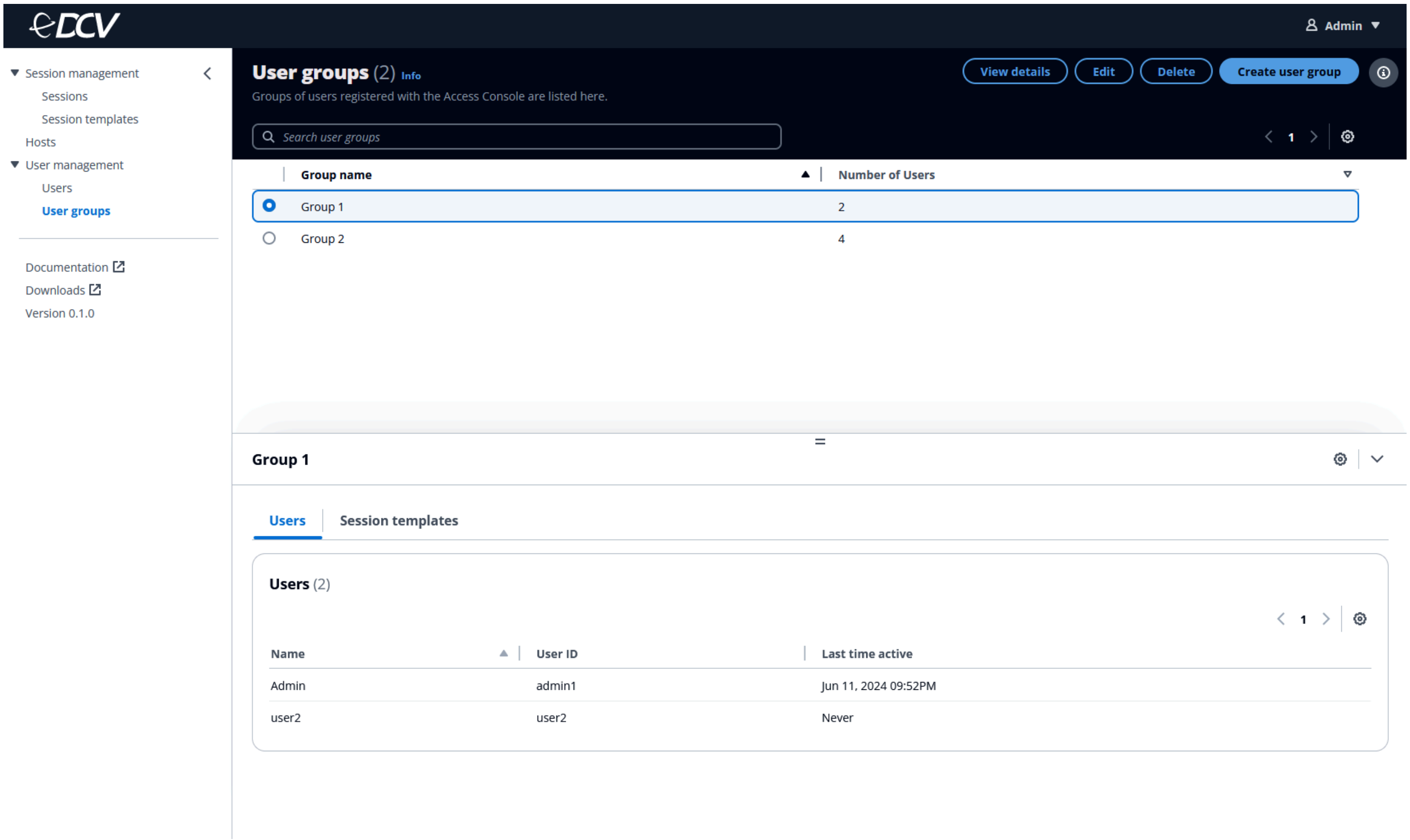
```
$ sudo systemctl restart dcv-access-console-handler
```

User groups

The Amazon DCV Access Console allows admins to manage user groups and their assigned templates. You can import user groups, or create and manage them from the Access Console itself.

On the **User groups** page, you can view the user groups you created or imported, and their detailed information. User groups can only include users that are saved in your datastore. For more information on how to import user groups, see Import users and groups.

User group details



User groups (2) Info

Groups of users registered with the Access Console are listed here.

	Group name	Number of Users
●	Group 1	2
○	Group 2	4

Users (2)

Name	User ID	Last time active
Admin	admin1	Jun 11, 2024 09:52PM
user2	user2	Never

Property	Description
Group name	The display name of the group.
Group ID	The unique ID of the group. This cannot be changed.
Number of users	The number of users assigned to the group.
Users	The users assigned to the group.
Session templates	The session templates assigned to the group.

Creating user groups

You can create a user group directly from the Access Console, by selecting users and assigning templates.

To create a user group

1. Go to the **User groups** page.
2. Select the **Create user group** button.
3. In **Group ID**, enter a unique identifier for your group. After you create the group, you cannot edit the ID.
4. In **Name**, enter a user friendly name for your group.
5. In **Add users**, select the users you wish to add to the group.
6. In **Template**, select the templates you wish to assign to the group.
7. Select the **Submit** button.

Editing user groups

You can edit a user group directly from the Access Console, and are able to modify the group name, users in the group and templates assigned to the group.

To edit a user group

1. Go to the **User groups** page.
2. Select the user group you want to edit.
3. Select the **Edit** button.
4. Edit the name, add or remove users, or add or remove session templates.
5. Select the **Submit** button.

Custom branding in the Amazon DCV Access Console

To create a familiar experience for your users when they Amazon DCV, you can customize the appearance of Amazon DCV Access Console with your own login background image, logo, login message, documentation links and client download links. When you customize the Amazon DCV Access Console, your branding is displayed to users rather than the default Amazon DCV branding.

Note

Any login background image, logo, login message, documentation links, and client download links you choose to use to customize the Amazon DCV Access Console is Your Content (as the term is defined in the Customer Agreement, which in turn is defined in the EULA). You are solely responsible for Your Content and your use of Your Content to customize the Amazon DCV Access Console, including compliance with the Policies as defined in the Customer Agreement and applicable law.

Custom branding options

You can customize the appearance of the Amazon DCV Access Console by using the following branding options.

Branding element	Description	Requirements and recommendations
Organization logo	Enables you to display an image that is familiar to your users. The image appears on the log in page, and at the top of the Console after the user has logged in (service-name.svg).	File type: .svg Recommended dimensions: 112 x 32 px
Favicon	Enables your users to recognize the Console site in a browser full of tabs or	File type: .ico Recommended dimensions: 28px x 28px

Branding element	Description	Requirements and recommendations
	bookmarks. The favicon icon is displayed at the top of the browser tab for the Console site (favicon.io).	
Login message	Enables you to customize a message on the log in screen.	Length constraints: Minimum length of 1 character. Maximum length of 200 characters. File type: .svg
Login background image	Enables you to customize the background image on the login screen (login-bacgroud.svg).	Recommended dimensions: 1440 px x 1024 px
Documentation URL	Enables you to specify a URL for a Documentation link.	Format: https://example.com or http://example.com
Downloads URL	Enables you to specify a URL for a Downloads link, so that users can download the appropriate native client to stream their Amazon DCV session from.	Format: https://example.com or http://example.com

Adding your custom branding

To customize the Amazon DCV Access Console with your organizational branding, you need to update the following with your preferred configurations:

- Authentication Server
- Web Client

Updating customization on the Authentication Server

1. Connect to the host on which you are running the Authentication Server.
2. Create a backup directory and copy the files that will be changed.

   ```
   $ mkdir custom_branding_bkp
   ```

   ```
   $ sudo cp /opt/aws/dcv-access-console-auth-server/dcv-access-console-auth-server-
   *.jar custom_branding_bkp/
   ```

3. Create a working directory.

   ```
   $ mkdir custom_branding
   ```

   ```
   $ cd custom_branding
   ```

4. Copy the Authentication Server.

   ```
   $ sudo cp /opt/aws/dcv-access-console-auth-server/dcv-access-console-auth-server-
   *.jar .
   ```

5. Unzip the relevant files.

   ```
   $ unzip dcv-access-console-auth-server-*.jar BOOT-INF/classes/static/_next/static/
   chunks/app/login/*.js
   ```

   ```
   $ unzip dcv-access-console-auth-server-*.jar BOOT-INF/classes/static/service-
   name.svg
   ```

   ```
   $ unzip dcv-access-console-auth-server-*.jar BOOT-INF/classes/static/favicon.ico
   ```

   ```
   $ unzip dcv-access-console-auth-server-*.jar BOOT-INF/classes/static/login-
   background.svg
   ```

6. Replace the existing images file paths with paths to your new custom **organization logo** , **favicon**, and **login background images**.

```
$ sudo cp path-to-new-favicon.ico BOOT-INF/classes/static/favicon.ico
```

```
$ sudo cp path-to-new-service-name.svg BOOT-INF/classes/static/service-name.svg
```

```
$ sudo cp path-to-new-login-background.svg BOOT-INF/classes/static/login-
background.svg
```

7. Update the **alternative text** for the organization logo.

```
$ OLD_ALT="Access Console"
```

```
$ NEW_ALT="My new logo alt text"
```

```
$ sudo sed -i "s/alt:\"$OLD_ALT\"/alt:\"$NEW_ALT\"/g" BOOT-INF/classes/static/
_next/static/chunks/app/login/page-*.js
```

8. Update the **login message** on the login screen.

```
$ OLD_TAGLINE="Manage and connect to your Amazon DCV sessions."
```

```
$ NEW_TAGLINE="My new tag line"
```

```
$ sudo sed -i "s/tagline:\"$OLD_TAGLINE\"/tagline:\"$NEW_TAGLINE\"/g" BOOT-INF/
classes/static/_next/static/chunks/app/login/page-*.js
```

9. Replace the files in the jar.

```
$ zip -ur dcv-access-console-auth-server-*.jar BOOT-INF/
```

10. Copy the new jar.

```
$ sudo cp dcv-access-console-auth-server-*.jar /opt/aws/dcv-access-console-auth-
server/
```

11. Reload the daemon and restart the authorization server.

```
$ sudo systemctl daemon-reload  sudo systemctl restart dcv-access-console-auth-
server
```

Updating customization on the Web Client

1. Connect to the host on which you are running the Web Client.
2. Create a backup directory and copy the files that will be changed.

   ```
   $ mkdir custom_branding_bkp
   ```

   ```
   $ sudo cp -r /opt/aws/dcv-access-console-webclient custom_branding_bkp/
   ```

3. Replace the existing images file paths with paths to your new custom **organization logo** , **favicon**, and **login background images** (the login background image is used on the Web Client for error messages).

   ```
   $ sudo cp path-to-new-service-name.svg /opt/aws/dcv-access-console-webclient/
   public/service-name.svg
   ```

   ```
   $ sudo cp path-to-new-favicon.ico.body/opt/aws/dcv-access-console-webclient/.next/
   server/app/favicon.ico.body
   ```

   ```
   $ sudo cp path-to-new-login-background.svg/opt/aws/dcv-access-console-webclient/
   public/login-background.svg
   ```

4. Update the **alternative text** for the organization logo.

   ```
   $ OLD_ALT="Access Console"
   ```

   ```
   $ NEW_ALT="My new logo alt text"
   ```

   ```
   $ grep -rl "alt:\"$OLD_ALT\"" /opt/aws/dcv-access-console-webclient/.next/ | xargs
    sed -i "s/alt:\"$OLD_ALT\"/alt:\"$NEW_ALT\"/g"
   ```

5. Replace the **Documentation** URL.

```
$ OLD_DOC_LINK="https:\/\/docs.aws.amazon.com\/dcv\/latest\/sm-admin\/what-is-
sm.html"
```

```
$ NEW_DOC_LINK="https:\/\/example.com"
```

```
$ grep -rl $OLD_DOC_LINK /opt/aws/dcv-access-console-webclient/.next/ | xargs sed -
i "s/$OLD_DOC_LINK/$NEW_DOC_LINK/g"
```

6. Replace the **Downloads** URL.

```
$ OLD_DOWNLOADS_LINK="https:\/\/download.nice-dcv.com\/"
```

```
$ NEW_DOWNLOADS_LINK="https:\/\/example.com"
```

```
$ grep -rl $OLD_DOWNLOADS_LINK /opt/aws/dcv-access-console-webclient/.next/ | xargs
 sed -i "s/$OLD_DOWNLOADS_LINK/$NEW_DOWNLOADS_LINK/g"
```

Configuration file reference

This section provides information about the Authentication Server, Handler and Web Client configuration files.

Topics

- Authentication Server configuration files
- Handler configuration files
- Web Client configuration files

Authentication Server configuration files

The Authentication Server has two configuration files (`/etc/dcv-access-console-auth-server/access-console-auth-server.properties` and `/etc/dcv-access-console-auth-server/access-console-auth-server-secrets.properties`) that include parameters that can be configured to customize the Amazon DCV Access Console functionality connecting to different components.

> **Note**
>
> The property files contains sensitive data. By default, its write access is restricted to root and its read access is restricted to root and to the user running the Authentication Server. By default, this is the `dcvaccessconsole` user.

The following tables list the parameters in the Authentication Server configuration files.

For the `/etc/dcv-access-console-auth-server/access-console-auth-server.properties` configuration:

Parameter name	Required	Default Value	Description
`server-port`	Yes	9000	Specifies the port the Authentication Server listens.

Parameter name	Required	Default Value	Description
authentication-header-name	Either authentication-header-name or pam-helper-path is required	username	Specifies the header name in the request to use as the userid.
pam-helper-path	Either authentication-header-name or pam-helper-path is required	/var/usr/dcv-access-console-auth-server/dcvpamhelper	Specifies the full path of the dcvpamhelper that is installed as part of the Authentication Server.
pam-service-name	Only required if pam-helper-path is specified	dcv	Specify 'dcv' if /etc/pam.d/dcv is installed or use system-auth on redhat based systems, common-auth on ubuntu/debian .
enable-pam-debug	Only required if pam-helper-path is specified	False	Enables or disables the debug logging for the dcvpamhelper .
pam-process-timeout	Only required if pam-helper-path is specified	10	Specifies the number of seconds to wait for the dcvpamhelper to finish.

Parameter name	Required	Default Value	Description
`pam-normalize-userid-enabled`	No	False	Enables or disables the use of `pam-normalize-userid-command` to normalize the different usernames to a userid.
`pam-normalize-userid-command`	No	id -u -nr	Specifies the command to use to normalize the username to a userid.
`redirect-uris`	Yes		Specifies the call back url of the Web Client. It should be of the format `https://webclient-host:webclient-port/api/auth/callback/dcv-access-console-auth-server` .
`post-logout-redirect-uris`	Yes		Specifies the url of the Web Client to redirect to after logout. It should be of the format `https://webclient-host:webclient-port` .

Parameter name	Required	Default Value	Description
`authorization-server-hostname`	Yes		Specifies the url of the Authentication Server. It should be of the format `https://auth-server-host:auth-server-port` .
`throttling-burst`	No	50	Specifies the bucket maximum capacity of the token bucket throttle algorithm.
`throttling-refill`	No	2	Specifies the bucket refill rate of the token bucket throttle algorithm.
`throttling-period-in-seconds`	No	1	Specifies the period in seconds for the bucket refill rate of the token bucket throttle algorithm.
`throttling-login-burst`	No	10	Specifies the bucket maximum capacity of the token bucket throttle algorithm for the `/login` endpoint.
`throttling-login-refill`	No	10	Specifies the bucket refill rate of the token bucket throttle algorithm for the `/login` endpoint.

Parameter name	Required	Default Value	Description
`throttling-login-period-in-seconds`	No	3600	Specifies the period in seconds for the bucket refill rate of the token bucket throttle algorithm for the `/login` endpoint.
`throttling-cache-max-size`	No	1000	Specifies the number unique IP address to track for throttling.
`throttling-cache-max-time-minutes`	No	20	Specifies the number minutes to track an IP address for throttlin g.
`access-token-time-to-live`	No	30s	Specifies the time to live for the access token.
`refresh-token-time-to-live`	No	2h	Specifies the time to live for the refresh token. It should be greater than the `access-token-time-to-live` .
`show-cookie-link`	No	FALSE	Enables or disables if a link to a privacy disclaimer shows on the sign in page.

Parameter name	Required	Default Value	Description
`cookie-link-target`	No		Specifies the link your users will be directed to for the privacy disclaimer. If you set `show-cookie-link` to false, leave it without a value.
`show-privacy-link`	No	FALSE	Enables or disables if a link to a privacy disclaimer shows on the sign in page.
`privacy-link-target`	No		Specifies the link your users will be directed to for the privacy disclaimer. If you set `show-privacy-link` to false, leave it without a value.

For the `/etc/dcv-access-console-auth-server/access-console-auth-server-secrets.properties` configuration:

Parameter name	Required	Default Value	Description
`ssl.enabled`	No	False	Enables SSL in Authentication Server.
`ssl.key-store-type`	No	PKCS12	Specifies the type of the Java Keystore file.
`ssl.key-store`	No		Specifies the path to the Java Keystore file.

Parameter name	Required	Default Value	Description
`ssl.key-store-password`	No		Specifies the password to the Java Keystore file.
`auth-server-client-id`	No	`dcv-access-console-web-client`	Specifies the client id for the Web Client. It should be the same in the Web Client pro perties.
`auth-server-client-secret`	No		Specifies the secret for the Web Client. It should be the same in the Web Client pro perties.

Handler configuration files

The Handler has two configuration files (`/etc/dcv-access-console-handler/access-console-handler.properties` and `/etc/dcv-access-console-handler/access-console-handler-secrets.properties`) that include parameters that can be configured to customize the Amazon DCV Access Console functionality connecting to different components.

Note

The property files contains sensitive data. By default, its write access is restricted to root and its read access is restricted to root and to the user running the Handler. By default, this is the `dcvaccessconsole` user.

The following table lists the parameters in the Handler configuration files.

For the `/etc/dcv-access-console-handler/access-console-handler.properties` configuration:

Parameter name	Required	Default Value	Description
`server-port`	Yes	8080	Specifies the port the Handler listens.
`web-client-url`	Yes		Specifies the url of the Web Client. It should be of the format `https://webclient-host:webclient-port` .
`client-to-broker-connector-url`	Yes		Specifies the url of the Broker. It should be of the format `https://broker-host:client-to-broker-connector-https-port` .
`client-to-broker-connector-auth-url`	Yes		Specifies the authentication url of the Broker. By default it is `https://broker-host:client-to-broker-connector-https-port/oauth2/token` .
`client-to-broker-connection-verify-ssl`	Yes		Enables SSL certificate validation for the connection between the Handler and the Broker.

Parameter name	Required	Default Value	Description
`enable-connection-gateway`	No		Enables the use of connection gateway to connect to the Amazon DCV server.
`connection-gateway-host`	Only required if enable-connection-gateway is true		Specifies the connection gateway host name to use while creating the connection url.
`connection-gateway-port`	Only required if enable-connection-gateway is true		Specifies the connection gateway port to use while creating the connection url.
`enable-public-ip-from-tag`	No		Enables the use of the DCV server tag to obtain the host name or IP address rather than the public DNS of the server.
`public-ip-tag-name`	Only required if `enable-public-ip-from-tag` is true		Specifies the tag name to use to obtain the host name or IP address.
`persistence-db`	Yes		Specifies which database is used for persistence. The only supported values are: `dynamodb` and `mysql`.

Parameter name	Required	Default Value	Description
`table-name-prefix`	No	`dcv_access_console_`	Specifies the prefix that is added to each table (useful to distinguish multiple Handler using the same AWS account). Only alphanumeric characters, dot, dash and underscore are allowed.
`persistence-db-default-max-results`	No	20	Specifies the maximum number of items to retrieve from database.

Parameter name	Required	Default Value	Description
jdbc-connection-url	Only required if persistence-db is set to mysql		Specifies the connection URL to the MariaDB/MySQL database; it contains the endpoint and the database name. The url should have this format:jdbc:mysql://*db_endpoint* :*db_port*/*db_name* *db_endpoint* is the MariaDB/MySQL database endpoint, *db_port* is the database port and *db_name* is the database name.
jpa-db-platform	Only required if persistence-db is set to mysql	org.hibernate.dialect.MariaDBDialect	Specifies the name of the target database.
dynamodb-region	Only required if persistence-db is set to dynamodb		Specifies the region where the DynamoDB tables are created and accessed.
request-prefix	No	/accessconsolehandler	Specifies the prefix for the Handler endpoints.

Parameter name	Required	Default Value	Description
`jwt-issuer-uri`	Yes		Specifies the Authentication Server URL. It is of the format https://auther-server-host:auth-server-port.
`user-id-case-sensitive`	No	True	Specifies if the userid should be case sensitive.
`authorization-policies-location`	Yes	`/etc/dcv-access-console-handler/authorization/policies.cedar`	Specifies the path to the Cedar policy file.
`authorization-roles-location`	Yes	`/etc/dcv-access-console-handler/authorization/roles.json`	Specifies the path to the Cedar roles file.
`default-role`	Yes		Specifies the default role to assign to new users.
`users-batch-save-size`	No	100	Specifies the number of users to save at a time to the database during user import.

Parameter name	Required	Default Value	Description
`import-users-cache-size`	No	1000	Specifies the number of users to keep in memory during user import to check if the user already exists.
`throttling-burst`	No	50	Specifies the bucket maximum capacity of the token bucket throttle algorithm.
`throttling-refill`	No	2	Specifies the bucket refill rate of the token bucket throttle algorithm.
`throttling-period-in-seconds`	No	1	Specifies the period in seconds for the bucket refill rate of the token bucket throttle algorithm.
`throttling-cache-max-size`	No	1000	Specifies the number unique IP address to track for throttling.
`throttling-cache-max-time-minutes`	No	20	Specifies the number minutes to track an IP address for throttling.

Parameter name	Required	Default Value	Description
`jwt-login-username-claim-key`	No		Specifies the key to retrieve login username from the JWT claims of the auth server.
`jwt-display-name-claim-key`	No		Specifies the key to retrieve display name from the JWT claims of the auth server.

Parameter name	Required	Default Value	Description
`jwt-default-groups-claim-key`	No		Specifies the key that contains the default groups. To enable access to session templates on first login, create groups and assign them session templates through the Access Console, then use the group IDs in your external OAuth configuration. Groups are only assigned on first login and subsequent group management should be done through the Access Console. The groups value should be comma-separated group IDs (e.g., "group1,group2"). If a group does not exist, it will be created.

Parameter name	Required	Default Value	Description
`jwt-role-claim-key`	No		Specifies the key to retrieve the role from the JWT claims of the auth server. The role value must match a configured role (e.g., Admin, User, Guest). Invalid roles fall back to the default role defined in `default-role` in the Handler configuration file.
`auth-server-well-known-uri`	No		Specifies the well known URI of the auth server.
`auth-server-userinfo-endpoint`	No		Specifies the userInfo endpoint of the auth server.
`auth-server-claims-from-access-token`	No	False	Specifies the userInfo endpoint of the auth server.

For the `/etc/dcv-access-console-handler/access-console-handler-secrets.properties` configuration:

Parameter name	Required	Default Value	Description
`ssl.enabled`	No	False	Enables SSL in Authentication Server.

Parameter name	Required	Default Value	Description
`ssl.key-store-type`	No	PKCS12	Specifies the type of the Java Keystore file.
`ssl.key-store`	No		Specifies the path to the Java Keystore file.
`ssl.key-store-password`	No		Specifies the password to the Java Keystore file.
`broker-client-id`	Yes		Specifies the client id to use for the Broker API calls.
`broker-client-password`	Yes		Specifies the client secret to use for the Broker API calls.
`jdbc-user`	Only required if persistence-db is set to `mysql`		Specifies the name of the user that has access to the MariaDB/MySQL database.
`jdbc-password`	Only required if persistence-db is set to `mysql`		Specifies the password of the user that has access to the MariaDB/MySQL database.

Web Client configuration files

The Web Client configuration has two configuration files (`/etc/dcv-access-console-webclient/access-console-webclient.properties` and `/etc/dcv-access-console-webclient/access-console-webclient-secrets.properties`) that include parameters that can be configured to customize the Amazon DCV Access Console functionality connecting to different components.

> **Note**
>
> The property files contains sensitive data. By default, its write access is restricted to root and its read access is restricted to root and to the user running the Web Client. By default, this is the `dcvaccessconsole` user.

The following tables list the parameters in the Web Client configuration files.

For the `/etc/dcv-access-console-webclient/access-console-webclient.properties` configuration:

Parameter name	Required	Default Value	Description
`server-port`	Yes	8080	Specifies the port to which the Handler listens
`web-client-url`	Yes		Specifies the url of the Web Client. It should be of the format `https://webclient-host:webclient-port` .
`client-to-broker-connector-url`	Yes		Specifies the url of the Broker. It should be of the format `https://broker-host:client-to-broker-connector-https-port` .

Parameter name	Required	Default Value	Description
`web-client-url`	Yes		Specifies the url of the Web Client. It should be of the format https://webclient-host:webclient-port.
`enable-connection-gateway`	No		Enables the use of connection gateway to connect to the Amazon DCV server.
`extra-ca-certs`	No		Specifies the path to a well known CA certificates in PEM format. If you followed the documentation to create a self signed certificate, then the value will be the path to `rootCA.pem` .

Parameter name	Required	Default Value	Description
`session-screenshot-max-width`	No	1280	Specifies the maximum pixel width of session screenshots taken using the GetSessionScreenshots API. This takes precedence over the values in the Session Manager Broker configuration file. If not specified , the default value will be used. If set to 0, the values from the Session Manger Broker configuration will apply.

Parameter name	Required	Default Value	Description
`session-screenshot-max-height`	No	960	Specifies the maximum pixel height of session screenshots taken using the GetSessionScreenshots API. This takes precedence over the values in the Session Manager Broker configuration file. If not specified, the default value will be used. If set to 0, the values from the Session Manger Broker configuration will apply.
`auth-server-scope`	Yes	openid	When using an external auth provider the custom scope can be set. Multiple scopes can be specified by separating them with spaces.

For the `/etc/dcv-access-console-webclient/access-console-webclient-secrets.properties` configuration:

Parameter name	Required	Default Value	Description
`auth-server-client-id`	Yes	`dcv-access-console-web-client`	Specifies the client id for the Web Client. It should be the same in the Authentication Server properties.
`auth-server-client-secret`	Yes		Specifies the secret for the Web Client. It should be the same in the Authentication Server properties.
`cookie-secret`	Yes		Specifies a random string used to sign/encrypt cookies and JWT.

Upgrading the Access Console

The following section explains how to update Amazon DCV Access Console components on a single host and on separate multiple hosts.

Topics

- Upgrading Amazon DCV Access Console on a single host
- Upgrading Amazon DCV Access Console on multiple hosts

Upgrading Amazon DCV Access Console on a single host

The Wizard will update the components for the Access Console, reload and restart all of the Access Console components. The components can be downloaded using steps in Preparing the components and the Setup Wizard.

Running the Setup Wizard in interactive mode

Interactive mode is the default update mode for the Amazon DCV Access Console. It will guide you through the update process.

1. Navigate to the folder where you extracted the latest Amazon DCV Access Console components.
2. Run the following command:

   ```
   $  python3 wizard.py update
   ```

3. Provide the path to the folder where the installers for the three components can be found. By default, it looks in the current directory.

The Wizard will first validate the processes are running, update them, reload and restart the Amazon DCV Access Console components.

Running the Setup Wizard in non-interactive mode

Non-interactive mode of the update wizard will allow for it be used in scripts.

1. Navigate to the folder where you extracted the latest Amazon DCV Access Console components.
2. Run the following command:

   ```
   $  python3 wizard.py update --component-installers-location
   ```

The Wizard will first validate the processes are running, update them, reload and restart the Amazon DCV Access Console components.

Upgrading Amazon DCV Access Console on multiple hosts

To upgrade the Handler, Authentication Server, and Web Client components, you must run the following commands. The components can be downloaded and extracted using the steps in Prepare your environment.

> **Note**
>
> These components need to be downloaded to each host being used.

Upgrading the Handler

RHEL, CentOS, Amazon Linux

1. Connect to the host you set up for the Handler.
2. Move the Handler `.rpm` file you downloaded to the host.
3. Stop the running service.

   ```
   $  sudo systemctl stop dcv-access-console-handler
   ```

4. Upgrade the Handler component.

   ```
   $  sudo yum install -y nice-dcv-access-console-handler*.rpm
   ```

5. Start the Handler component.

   ```
   $  sudo systemctl daemon-reload
   ```

```
$  sudo systemctl restart dcv-access-console-handler
```

Ubuntu, Debian

1. Connect to the host you set up for the Handler.
2. Move the Handler `.deb` file you downloaded to the host
3. Stop the running service.

   ```
   $  sudo systemctl stop dcv-access-console-handler
   ```

4. Upgrade the Handler component.

   ```
   $  sudo apt install -y ./nice-dcv-access-console-handler*.deb
   ```

5. Start the Handler component.

   ```
   $  sudo systemctl daemon-reload
   ```

   ```
   $  sudo systemctl restart dcv-access-console-handler
   ```

Upgrading the Authentication Server

RHEL, CentOS, Amazon Linux

1. Connect to the host you set up for the Authentication Server.
2. Move the Authentication Server `.rpm` you downloaded to the host.
3. Stop the running service.

   ```
   $  sudo systemctl stop dcv-access-console-auth-server
   ```

4. Upgrade the Authentication Server component.

   ```
   $  sudo yum install -y nice-dcv-access-console-auth-server*.rpm
   ```

5. Start the Authentication Server component.

```
$ sudo systemctl daemon-reload
```

```
$ sudo systemctl restart dcv-access-console-auth-server
```

Ubuntu, Debian

1. Connect to the host you set up for the Authentication Server.
2. Move the Authentication Server `.deb` you downloaded to the host.
3. Stop the running service.

   ```
   $ sudo systemctl stop dcv-access-console-auth-server
   ```

4. Upgrade the Authentication Server component.

   ```
   $ sudo apt install -y ./nice-dcv-access-console-auth-server*.deb
   ```

5. Start the Authentication Server component.

   ```
   $ sudo systemctl daemon-reload
   ```

   ```
   $ sudo systemctl restart dcv-access-console-auth-server
   ```

Upgrading the Web Client

RHEL, CentOS, Amazon Linux

1. Connect to the host you set up for the Web Client.
2. Move the Web Client `.rpm` you downloaded to the host.
3. Stop the running service.

   ```
   $ sudo systemctl stop dcv-access-console-web-client
   ```

4. Upgrade Web Client component.

   ```
   $ sudo yum install -y nice-dcv-access-console-web-client*.rpm
   ```

5. Start the Web Client.

   ```
   $ sudo systemctl daemon-reload
   ```

   ```
   $ sudo systemctl restart dcv-access-console-web-client
   ```

Ubuntu, Debian

1. Connect to the host you set up for the Web Client.
2. Move the Web Client `.deb` you downloaded to the host.
3. Stop the running service.

   ```
   $ sudo systemctl stop dcv-access-console-web-client
   ```

4. Uninstall the existing Web Client component.

   ```
   $ sudo apt remove -y nice-dcv-access-console-web-client
   ```

5. Upgrade the Web Client component.

   ```
   $ sudo apt install -y ./nice-dcv-access-console-web-client*.deb
   ```

6. Start the Web Client.

   ```
   $ sudo systemctl daemon-reload
   ```

   ```
   $ sudo systemctl restart dcv-access-console-web-client
   ```

Troubleshooting

This section explains how to identify and troubleshoot problems that you might have with Amazon DCV Access Console.

There are a number of tools that Amazon DCV provides to help you in identifying any issues that occur with the Amazon DCV Access Console. You can use any of the following methods to help you identify possible problems.

Topics

Using the component log files

You can use the Amazon DCV Access Console component log files to identify and troubleshoot problems with the different Amazon DCV Access Console components. The component logs contain information about requests, responses, and errors regarding the component. The component access log files contain information about access, throttling, authentication, and authorization.

The log files can be found in the following locations on the host server that the Amazon DCV components are running on:

- Authentication Server

  `/var/log/dcv-access-console-auth-server/DCV-access-console-auth-server.log`

  `/var/log/dcv-access-console-auth-server/DCV-access-console-auth-server-access.log`
- Handler

`/var/log/dcv-access-console-handler/DCV-access-console-handler.log`

`/var/log/dcv-access-console-handler/DCV-access-console-handler-access.log`

- Web Client

  `/var/log/dcv-access-console-webclient/DCV-access-console-webclient.log`

  `/var/log/dcv-access-console-webclient/DCV-access-console-webclient-access.log`
- Ngnix

  `/var/log/nginx/error.log`

  `/var/log/nginx/access.log`

The Amazon DCV Access Console components enable you to configure the verbosity level of the log files. The following verbosity levels are available:

- `error` – Provides the least detail. Includes errors only.
- `warn` – Includes errors and warnings.
- `info` – The default verbosity level. Includes errors, warnings, and information messages.
- `debug` – Provides the most detail. Provides detailed information that is useful for debugging issues.

If you need to locate the logs for the session manager broker or session manager agent, see Amazon DCV Session Manager administrator guide.

Changing log file verbosity

To configure the log file verbosity, you must configure the log setting file by updating the `logback.xml` file with the appropriate class names and then restart the component processes.

Changing the Authentication Server log file verbosity

1. Navigate to `/etc/dcv-access-console-auth-server` and open the `logback.xml` file with your preferred text editor.
2. Update the level for `com.amazon.dcv.sm.ui` to the desired level of verbosity.

3. Update the level for `com.amazon.dcv.sm.ui.authserver.throttling` to the desired level of verbosity.

To change the Handler log file verbosity

1. Navigate to `/etc/dcv-access-console-handler` and open the `logback-spring.xml` file with your preferred text editor.
2. Update the level for `com.amazon.dcv.sm.ui` to the desired level of verbosity.
3. Update the level for `com.amazon.dcv.sm.ui.handler.authorization` to the desired level of verbosity.
4. Update the level for `com.amazon.dcv.sm.ui.authserver.throttling` to the desired level of verbosity.

Using browser and network log files

The web browser communicates with the Handler component to view and modify resources. If there are issues with communication between the web browser and the Handler, you can troubleshoot using the browser and network log files.

Accessing Chrome console logs

From a Chrome browser, access the console log window.

1. Do one of the following:
   - Use the shortcut key. For Windows and Linux, use `Ctrl+Shift+J`. For macOS, use, `Cmd+Opt+J`.
   - Select the Chrome menu button on the upper right hand side, select **More Tools** then choose **Developer Tools**.
2. Select the **Console** tab in the **Developer Tools** pane.

In the **Console** tab, errors are highlight in red and warnings are highlight in yellow.

Accessing Chrome network logs

From a Chrome browser, the network tab contains network calls for uploaded and downloaded resources.

1. Do one of the following:

   - Use the shortcut key. For Windows and Linux, use `Ctrl+Shift+J`. For macOS, use, `Cmd+Opt+J`.
   - Select the Chrome menu button on the upper right hand side, select **More Tools** then choose **Developer Tools**.

2. Select the **Network** tab in the **Developer Tools** pane.
3. Refresh the page.

Errors are highlighted in red. Select an error to see more information about it.

The **Status Code** in both the **Headers** and the **Response** tabs can be used to diagnose issues.

Managing the component processes

The Amazon DCV Access Console components, such as Authentication Server, Handler, Web Client, run while processes on their hosts and can be managed using the command `systemctl`. You can use this command to:

- Check the status of a component
- Stop a component
- Start a component
- Restart a component

If your components are running on separate hosts, then each command must be executed on each corresponding host.

Checking status of the components

To check the statuses of the components, run the following commands on the hosts that the components are installed on.

```
sudo systemctl status dcv-access-console-auth-server
sudo systemctl status dcv-access-console-handler
sudo systemctl status dcv-access-console-webclient
```

Stopping the components

To stop the component processes, run the following commands on the hosts that the components are installed on.

```
sudo systemctl stop dcv-access-console-auth-server
sudo systemctl stop dcv-access-console-handler
sudo systemctl stop dcv-access-console-webclient
```

Starting the components

To start the component processes, run the following commands on the hosts that the components are installed on.

```
sudo systemctl start dcv-access-console-auth-server
sudo systemctl start dcv-access-console-handler
sudo systemctl start dcv-access-console-webclient
```

Restarting the components

To restart the component processes, run the following commands on the hosts that the components are installed on.

```
sudo systemctl restart dcv-access-console-auth-server
sudo systemctl restart dcv-access-console-handler
sudo systemctl restart dcv-access-console-webclient
```

Handler fails to communicate with the broker

If there are communication failures between Handler component and Session Manager Broker, “Broker authentication error” will appear in the browser logs or `BrokerAuthenticationException: {"error":"unauthorized_client"}` in the handler logs. This is due to the fact that the Broker has incorrect property files or the Handler is unable to connect to the Broker.

Incorrect Broker properties

The Handler communicates with the Session Manager Broker using the properties specified in the `session-manager-handler.properties` file. If the property files are incorrect, communication issues can occur between the two.

1. On the host where the Handler is installed, navigate to the Handler properties file using your preferred text editor.

   ```
   /etc/dcv-access-console-handler/access-console-handler.properties
   ```

2. Verify that the `broker-base-url points to the Broker URL with the client-to-broker-connector-https-port`. For more information, see Broker configuration file in the *Amazon DCV administrator guide*.
3. Verify that the `broker-auth-url` points to the Broker authentication URL.
4. Verify that the `broker-client-id` and `broker-client-password` are correct. If you do not know the client-id and password you can register a new client using the `register-api-client` broker api.
5. Restart the Handler.

   ```
   sudo systemctl restart dcv-access-console-handler
   ```

Handler is unable to connect to the Broker

The Handler needs to connect to the Session Manager Broker on the `client-to-broker-connector-https-port` of the Broker. To verify that the Handler can connect to the Broker, run telnet to the Broker host name and the `client-to-broker-connector-https-port` (8443 by default) on the host where the Handler is installed.

If you are unable to connect to the host where the Broker is installed, see Networking and connectivity for requirements.

Example of a successful connection:

```
telnet broker-host 8443
Trying broker-host ip address...
Connected to broker-host.
```

```
Escape character is '^]'.
^]
telnet> ^C
```

I'm having problems logging in

During login, the Web Client uses OAuth 2.0 with the Authentication Server to receive an access token that is used to obtain user information and other information from the Handler. If you experience errors logging in, it could be due to either an error contacting the Handler, or invalid PAM credentials if you configured your Console to use PAM.

Error contacting the Handler

If you see an “Error contacting the handler” message, this means that the Web Client is unable to contact the Handler.

1. Check the status of the handler and the handler components logs to diagnose the problem.
2. Check that the web browser is able to connect to the host running the Handler. You could do this by using telnet to test connectivity to the port.

```
telnet handler-host 443
Trying handler-host ip address...
Connected to handler-host.
Escape character is '^]'.
^]
telnet> ^C
```

Invalid PAM credentials

When the Authentication Server is setup to use PAM authentication, it validates the username and the password using the PAM method of the operating system on the host running the authentication server.

Verify PAM authentication configuration

1. Connect to the host on which you are running the Authentication Server.
2. Navigate to `/etc/dcv-access-console-auth-server/access-console-auth-server.properties` .

3. Verify that `pam-service-name` is set to `system-auth` for Red Hat based systems or `common-auth` for Ubuntu/Debian.
4. Restart the Authentication Server.

Gather more detailed information.

1. Connect to the host on which you are running the Authentication Server.
2. Navigate to `/etc/dcv-access-console-auth-server/access-console-auth-server.properties`.
3. Enable `pam-normalize-userid-enabled` to true.
4. Enable debug logs for the `com.amazon.dcv.sm.ui.handler.authorization` class.
5. Restart the Authentication Server.

> **Note**
>
> Enabling “Debug” logging prints the access and refreshes tokens in the logs. It is recommended you change the verbosity back to “Info” after debugging.

Known issues

The Amazon DCV Access Console has the following known issues.

Cannot delete users from UI

To prevent users from logging into the UI, users can be disabled. To disable users, import the users with the `disabled` column set to true for the user.

Cannot manage Amazon DCV host servers

While the Access Console allows administrators to view the underlying hosts they have the Amazon DCV sessions installed on. However, it does not allow administrators to manage those resources directly. If you wish to start, terminate, or reboot your hosts, you must do so from your cloud or on-premise environment directly.

Security

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon DCV, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Amazon DCV. The following topics show you how to configure Amazon DCV to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon DCV resources.

Topics

Data protection in Amazon DCV

The AWS shared responsibility model applies to data protection in Amazon DCV. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Amazon DCV or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Data encryption

A key feature of any secure service is that information is encrypted when it is not being actively used.

Encryption at rest

Amazon DCV does not itself store any customer data. Data on Amazon DCV Server host can be encrypted at rest. When using Amazon DCV on AWS, please refer to the Encryption at rest section in the *Amazon EC2 User Guide* and to the Encryption at rest section in the *Amazon EC2 User Guide*.

Encryption in transit

All data transmitted from the Amazon DCV Client and Amazon DCV Server is encrypted by sending everything through a HTTPS/TLS connection.

To configure the certificates refer Managing the TLS certificate.

Compliance validation for Amazon DCV

Third-party auditors assess the security and compliance of AWS services as part of multiple AWS compliance programs. Using Amazon DCV to access a service does not alter that service's compliance.

For a list of AWS services in scope of specific compliance programs, see AWS services in scope by compliance program. For general information, see AWS compliance programs.

You can download third-party audit reports using the AWS Artifact. For more information, see Downloading reports in AWS Artifact.

Your compliance responsibility when using Amazon DCV is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and compliance quick start guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- AWS compliance resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating resources with rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub CSPM – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Release Notes and Document History for Amazon DCV Access Console

This page provides the release notes and document history for Amazon DCV Access Console.

Topics

Amazon DCV Access Console Release Notes

This section provides release notes for the Amazon DCV Access Console by release date.

Topics

2025.0-175 — February 2, 2026

Build numbers	Release notes
Version: 2025.0-175 - Web Client: 175	- Added `jwt-default-groups-claim-key` and `jwt-role-claim-key`

Build numbers	Release notes
• Handler: 175 • Authentication Server: 175 • Setup Wizard: 175	parameters in the Handler configuration to support role and group assignment from external OAuth claims. • Added `loginUsername` support to CSV user import. • Replaced `userId` values with `loginUsername` values for display throughout the user interface and added support for filtering and sorting on these values.

2025.0-168 — December 23, 2025

Build numbers	Release notes
Version: 2025.0-168 • Web Client: 168 • Handler: 168 • Authentication Server: 168 • Setup Wizard: 168	• Dropped support for Amazon Linux 2 due to security vulnerabilities in required packages.

2025.0-159 — November 12, 2025

Build numbers	Release notes
Version: 2025.0-159 • Web Client: 159 • Handler: 159 • Authentication Server: 159	• Added support for macOS hosts. •

Build numbers	Release notes
- Setup Wizard: 159	Fixed token expiration handling to support both seconds and milliseconds timestamp formats, preventing token expiration issues.

2025.0-155 — October 23, 2025

Build numbers	Release notes
Version: 2025.0-155 - Web Client: 155 - Handler: 155 - Authentication Server: 155 - Setup Wizard: 155	- Updated version to 2025.

2024.0-150 — June 17, 2025

Build numbers	Release notes
Version: 2024.0-150 - Web Client: 150 - Handler: 150 - Authentication Server: 150 - Setup Wizard: 150	- Added parameters in the Handler and Web Client configuration files to support external OAuth providers. - Other fixes and performance improvements.

2024.0-135 — January 15, 2025

Build numbers	Release notes

Build numbers	Release notes
Version: 2024.0-135 - Web Client: 135 - Handler: 94 - Authentication Server: 90 - Setup Wizard: 75	- Added configurable parameters in the Web Client configuration file to specify the maximum height and width of screenshots taken using the `GetSessionScreenshots` API. - Fixed an issue where session template requirements were not persisting when editing existing templates. - Fixed Web Client failing on EL9 based distributions. - Removed internet access requirement for Web Client installation. - Bug fixes and performance improvements.

2024.0-73 — October 1, 2024

Build numbers	Release notes
Version: 2024.0-73 - Web Client: 73 - Handler: 55 - Authentication Server: 54 - Setup Wizard: 50	- Rebranded NICE DCV to Amazon DCV. - Added support for Ubuntu 24.04. - Added functionality to make the Privacy link on the Sign In page configurable. - Bug fixes and performance improvements.

2023.1-57 — August 1, 2024

Build numbers	Release notes
Version: 2023.1-57	- Added the ability to upgrade the Access Console components in place.

Build numbers	Release notes
• Web Client: 57 • Handler: 39 • Authentication Server: 34 • Setup Wizard: 31	• Added the ability to select multiple session templates at once. • Modified the Setup Wizard to also be compatible with Python 3.6 and 3.7. • Bug fixes and performance improvements.

2023.1-20 — June 26, 2024

Build numbers	Release notes
Version: 2023.1-20 • Web Client: 20 • Handler: 20 • Authentication Server: 26 • Setup Wizard: 20	• Added an error if **Creating a session** fails. • Bug fixes and performance improvements.

2023.1 — June 13, 2024

Build numbers	Release notes
Version: 2023.1 • Web Client: 16 • Handler: 17 • Authentication Server: 25 • Setup Wizard: 15	Initial release of the Amazon DCV Access Console.

Document History

The following table describes the documentation for this release of Amazon DCV Access Console.

Change	Description	Date
Amazon DCV Version 2025.0-175	Amazon DCV Access Console has been updated for Amazon DCV 2025.0-175. For more information, see 2025.0-175--February 2, 2026.	February 2, 2026
Amazon DCV Version 2025.0-168	Amazon DCV Access Console has been updated for Amazon DCV 2025.0-168. For more information, see 2025.0-168--December 23, 2025.	December 23, 2025
Amazon DCV Version 2025.0-159	Amazon DCV Access Console has been updated for Amazon DCV 2025.0-159. For more information, see 2025.0-159--November 12, 2025.	November 12, 2025
Amazon DCV Version 2025.0-155	Amazon DCV Access Console has been updated for Amazon DCV 2025.0-155. For more information, see 2025.0-155--October 23, 2025.	October 23, 2025
Amazon DCV Version 2024.0-150	Amazon DCV Access Console has been updated for Amazon DCV 2024.0-150. For more information, see 2024.0-150--June 17, 2025.	June 17, 2025
Amazon DCV Version 2024.0-135	Amazon DCV Access Console has been updated for Amazon	January 15, 2025

Change	Description	Date
	DCV 2024.0-135. For more information, see 2024.0-135--January 15, 2025.	
Amazon DCV Version 2024.0-73	Amazon DCV Access Console has been updated for Amazon DCV 2024.0-73. For more information, see 2024.0-73--October 1, 2024.	October 1, 2024
Amazon DCV Version 2023.1-57	Amazon DCV Access Console has been updated for Amazon DCV 2023.1-57. For more information, see 2023.1-57--July 29, 2024.	August 1, 2024
Amazon DCV Version 2023.1-20	NICE DCV Access Console has been updated for NICE DCV 2023.1-20. For more information, see 2023.1-20--June 26, 2024.	June 26, 2024
Initial release	First publication of this content.	June 13, 2024