
PopcornSAR
Specialized in AUTOSAR
PopcornSAR Start Kit
AWS AMI Setting Guide

Popcornsar Software PopcornSAR Start kit V1.x.x - 10.2024

Privacy Policy for Personal Data Processing

Introduction
This privacy policy outlines how personal data is processed and protected when using our services or products. We are committed to ensuring that your privacy is protected in compliance with applicable data protection regulations, including the General Data Protection Regulation (GDPR).

1. Data Collection
We may collect and process the following personal data:

- **User Data**: Such as your name, username, and contact details.
- **Communication Data**: Including your IP address and other metadata for communication purposes.
- **Usage Data**: Information related to your interactions with our products or services.

2. Purpose of Data Processing
The personal data we collect is used for the following purposes:

- Providing and maintaining our services.
- Managing licenses and access to our products.
- Improving user experience through analytics.
- Compliance with legal obligations and protecting our rights.

3. Data Storage and Retention
Your personal data will be stored securely and will only be retained for as long as necessary to fulfill the purposes for which it was collected, or as required by law.

4. Data Security
We take appropriate technical and organizational measures to protect your personal data from unauthorized access, alteration, or disclosure. This includes:

- Encryption of sensitive data.
- Secure storage solutions.
- Regular data access audits and system monitoring.

5. Third-Party Sharing
We will not share your personal data with third parties without your consent, except when required by law or necessary to provide our services (e.g., with trusted service providers).

Contents

1 Product Introduction 5
2 Product license 5
3 Prerequisites and Requirements 6
4 AWS Account 6
5 AWS-Used Service 7
6 Technical prerequisites and requirements 7
7 Any prior technology or expertise required for deployment 7
8 Service Architecture 7
8-1 Public Resources 8
8-2 Elements 8
8-3 software components 9
8-4 architecture flow 9
8-5 summation 9
9 Deployment process 10
9-1 network settings 10
9-2 Account environment settings 10
10 Accessing the EC2 Instance 13
10-1 Default Username for the AMI 13
10-2 Access Methods 13
11 Actions when launching an EC2 instance 14
12 How to Connect via GUI (Graphical User Interface) 16
12 Apply a license to your product 19
13 Contact 20

1 Product Introduction

PopcornSAR is a development tool vendor specializing in the Adaptive AUTOSAR platform. Key products include AutoSAR.io, which designs systems and software according to the AUTOSAR standard, PARA, which implements the Adaptive AUTOSAR function cluster, and PACON IDE, an integrated development environment with various plug-ins.

AutoSAR.io and PARA described above are produced with AWS AMI to provide more convenient and diverse AutoSAR Adaptive to customers.

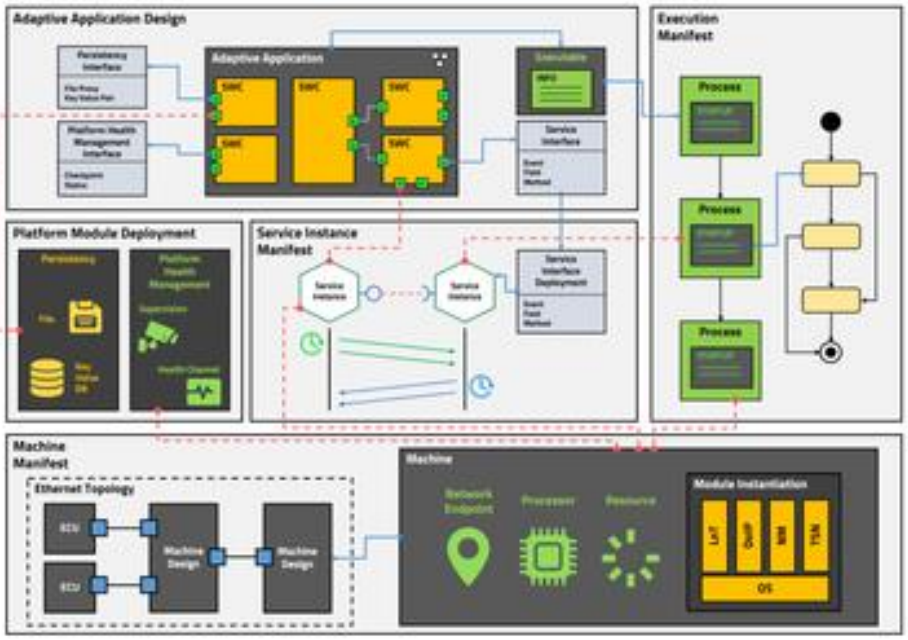
2 Product license

Pay as you go (PAYG) - Subscribe and pay through AWS Marketplace. Time-based usage; costs are based on AWS Marketplace and EC2 usage.

Bring Your Own License (BYOL) - Subscribe through AWS Marketplace and apply your existing license.

3 Prerequisites and Requirements

For PopocornSAR products, you must have a basic understanding of AutoSAR to use the product. See link: https://www.autosar.io/products/index



- PopcornSAR's Adaptive AutoSAR Application primarily uses the C++ programming language.
  - You must have basic technical knowledge of the C++ language.
- For this AMI product, an understanding of the services below provided by AWS is required.
  - AWS EC2, AMI
  - AWS Virtual Private Cloud (VPC)
    - A basic understanding of network communications is required.
    - Inbound, outbound communication
    - Understanding Internet Gateway (IGW)

4 AWS Account

- Use an account registered with AWS as an individual.

By default, IAM is not set, and IAM must be set to use NICE DCV.

5 AWS-Used Service

- Amazon Elastic Compute Cloud (Amazon EC2)
  - AMI
- Amazon Virutal Private Cloud (Amazon VPC)
  - subnet
  - security group
  - Internet Gateway
- Amazon NICE DCV
- AWS Secret Manager

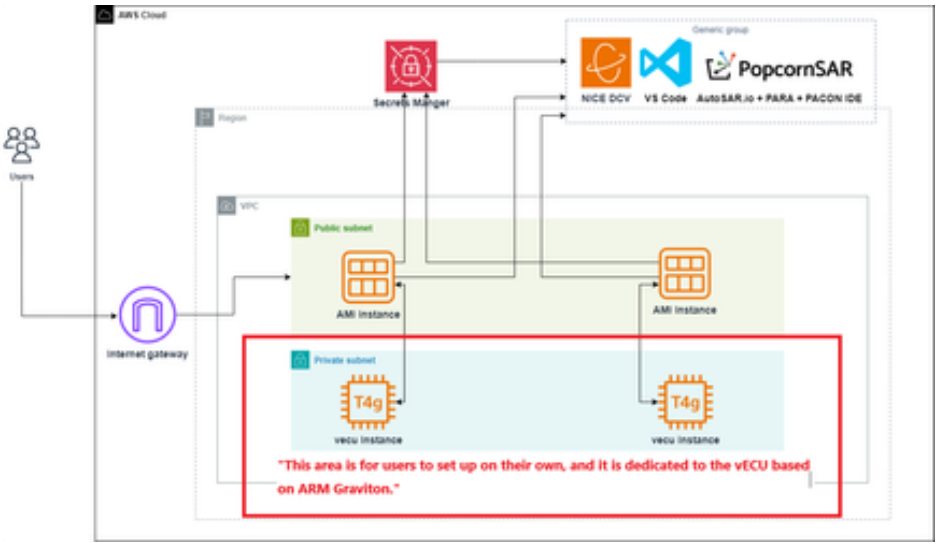
6 Technical prerequisites and requirements

- This AMI is version ubuntu22.04 and can only operate on x86_64.

7 Any prior technology or expertise required for deployment

- A good understanding of AWS VPC deployment is required.
- A good understanding of AWS EC2 and ssh connections is required.
- To use AutoSAR.io, an understanding of the C++ language is required.

8 Service Architecture



8-1 Public Resources

Public resources are resources that can be accessed directly from the Internet. Resources placed in public subnets are considered public resources.

1. Internet Gateway: Connects the VPC to the external Internet.
2. Public Subnet: A subnet that can be directly connected to the Internet.
3. AMI Instance: An EC2 instance that is placed in a public subnet and can communicate directly with the internet.

8-2 Elements

1. **Internet Gateway**
    1. It serves to connect traffic between the Internet and the AWS Cloud.
2. **Virtual Private Cloud**
    1. Form virtual networks to isolate applications and increase security.
3. **Availability Zones**
    1. Availability Zones 2a and 2c represent physically separate data centers, with instances distributed across multiple Availability Zones for high availability.
4. **Subnets**
    1. **Public Subnet:** A subnet that has direct access to the Internet. This is where the AMI instances are deployed.
    2. **Private Subnet:** A subnet that is not directly accessible from the Internet. This is where the T4g vCPU instances are deployed.
5. **AMI instance**
    1. An EC2 instance located in a public subnet that can communicate directly with the internet.
6. **T4g vCPU instance**
    1. Instances located in a private subnet communicate with instances in the public subnet, but do not communicate directly with the internet.
7. **Secrets Manager**
    1. Role: An AWS service that securely stores and manages secret values (e.g. API keys, database credentials, etc.).
    2. Function: Maintains security by providing secret values required by each application (NICE DCV, VS Code, PopcornSAR, etc.).

8-3 software components

1. **NICE DCV, VS Code**
    1. Software for remote desktop and code editing.
2. **PopcornSAR**
    1. This is the main application of the system. Includes submodules such as Auto-SARLab, RABA, PACON IDE, etc.

8-4 architecture flow

- The internet gateway allows external users to access AMI instances in the public subnet.
- AMI instances in the public subnet communicate with T4g vCPU instances in the private subnet as needed.
- Secrets Manager securely provides the secret values needed by many applications (NICE DCV, VS Code, PopcornSAR, etc.).
- This architecture is designed with high availability and security in mind, and distributes instances across multiple Availability Zones to minimize service interruption in the event of a failure.

8-5 summation

This architecture is designed to help you efficiently deploy and manage applications in the AWS cloud environment. Maintain high availability and security by leveraging internet gateways, public and private subnets, and Availability Zones. Use Secrets Manager to securely manage secret values and enable secure communication between applications.

9 Deployment process

1. network settings

- Set VPC and network information to use the AMI product as an EC2 instance.

VPC settings are set up by referring to the AWS official guidelines below.

- Link : https://docs.aws.amazon.com/vpc/latest/userguide/create-vpc.html

🚨 **Caution** 🚨 The following conditions are basically required to share product AMI.

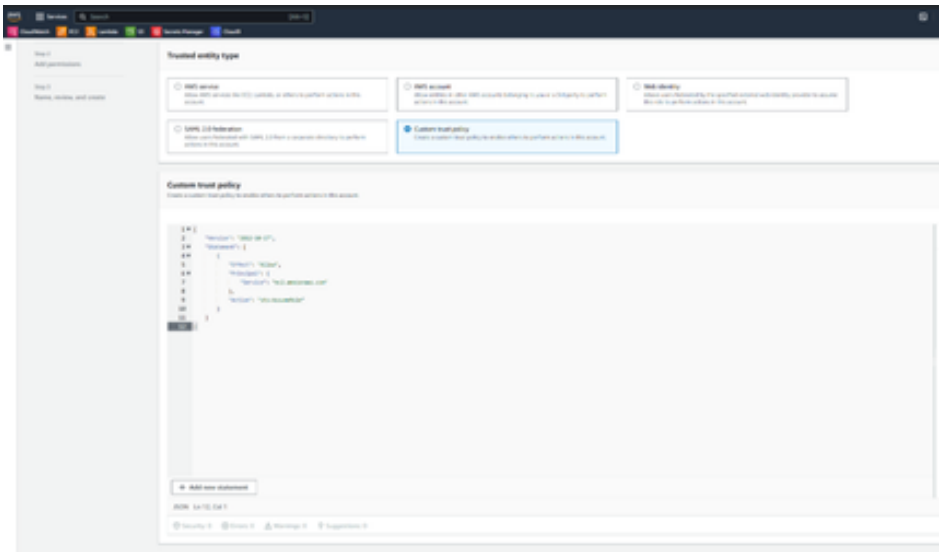
- Shared AMI region = Shared AMI region
  - Shares apply in the same region by default.

2. Account environment settings

The process of obtaining AMI from the marketplace is omitted.

2-1 IAM service access

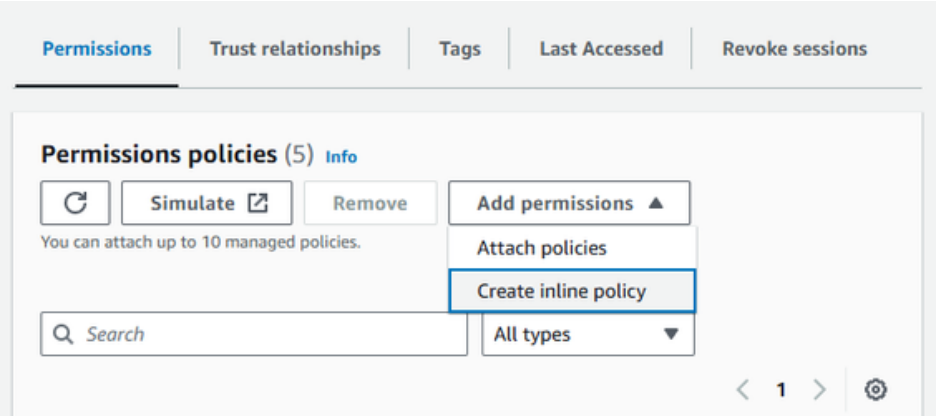
2-2 Create an IAM role



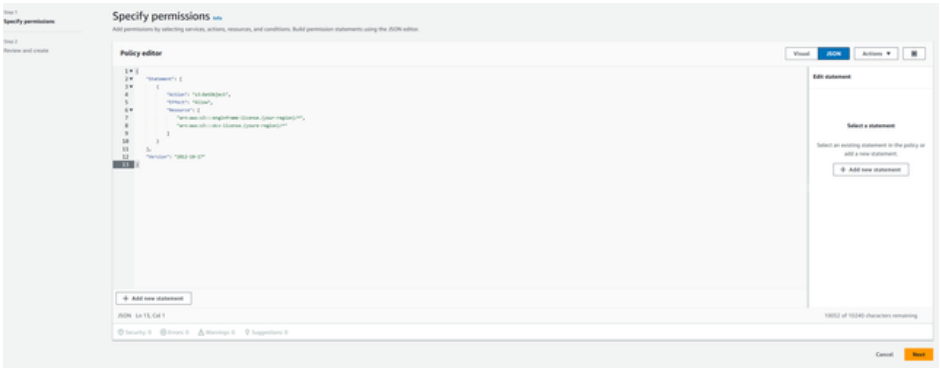
3. **Create a trust relationship for the role**

```
{
 "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "http://ec2.amazonaws.com  "
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

4. **Link policies**
    1. **Create required policies**



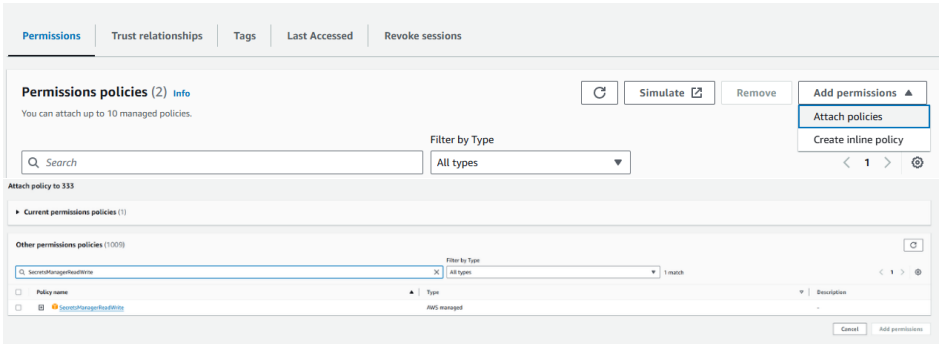
2. dcv license connection



5. Connect the policy by modifying {your-region} to fit your region

```
{
    "Statement": [
        {
            "Action": "s3:GetObject",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::enginframe-license.{your-region}/*",
                "arn:aws:s3:::dcv-license.{youre-region}/*"
            ]
        }
    ],
    "Version": "2012-10-17"
}
```

6. SecretsManagerReadWrite policy attachment



10 Accessing the EC2 Instance

This section explains how to access the EC2 instance created using the AMI

1. Default Username for the AMI

- **The EC2 instance created with this AMI can only be accessed using the username "ubuntu"**
- **Always use the "ubuntu" account when accessing via SSH or remote desktop**.

2. Access Methods

- There are two ways to access the EC2 instance:

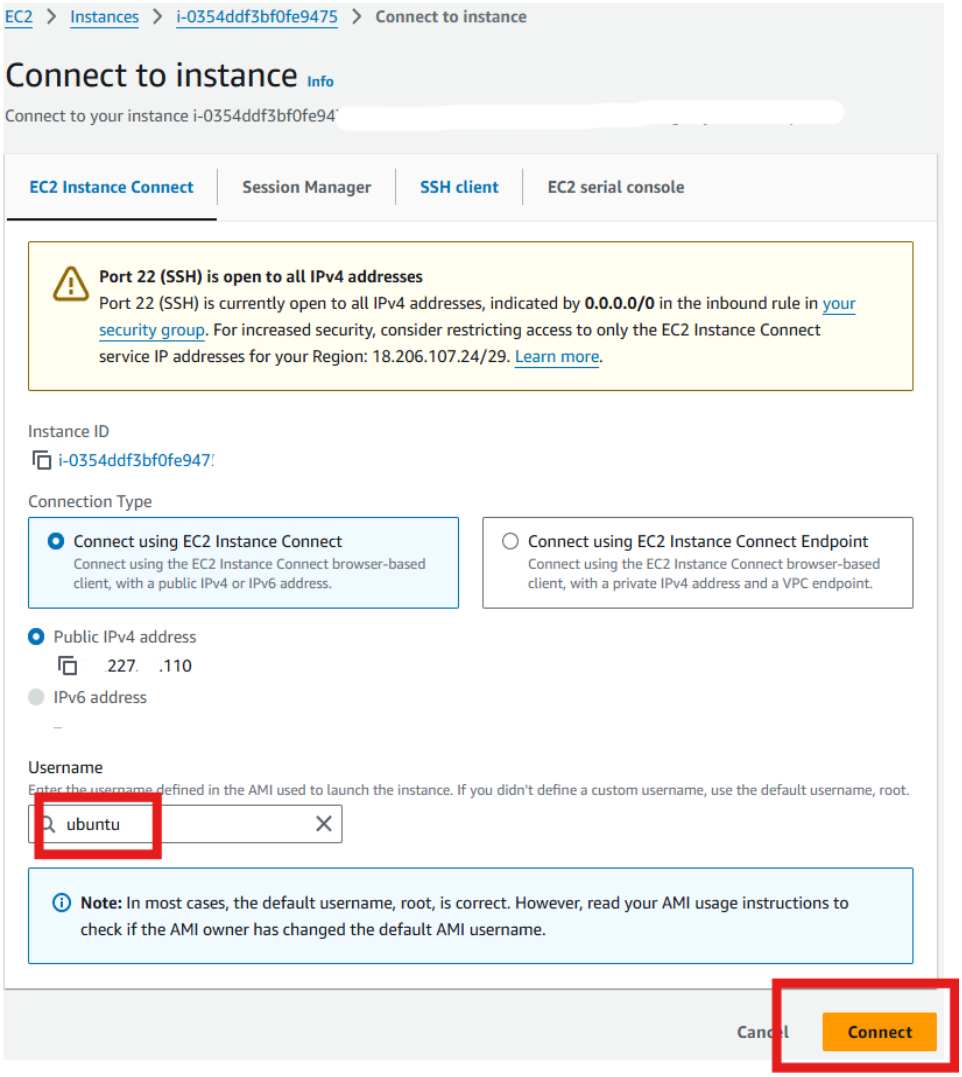
- Method 1: Access via GUI, such as NICE DCV
  - To access the instance using a graphical user interface, follow the instructions in section 11, where you connect to the instance using its public DNS.
- Method 2: Access via SSH using a key file
  - To access via SSH, you need the key file (.pem) specified during the instance creation.
  - Use the following command to connect via SSH:

    ssh -i /path/to/your-key-file.pem ubuntu@your-instance-public-dns

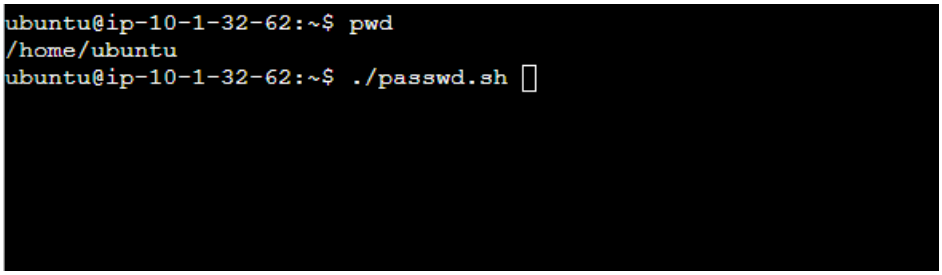
  - Replace your-key-file.pem with the path to your key file and your-instance-public-dns with the instance's public DNS address.

11 Actions when launching an EC2 instance

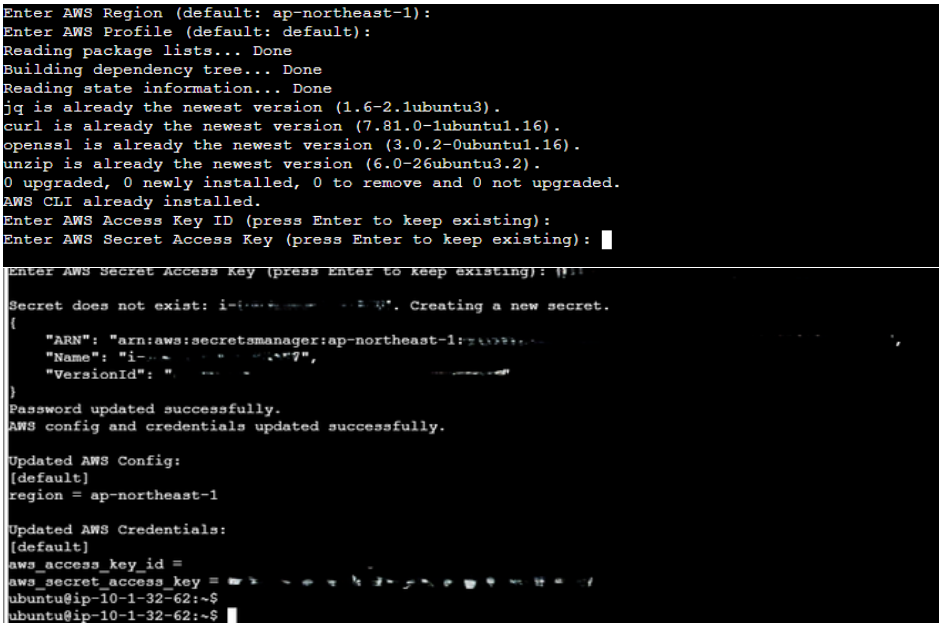
1. Click connect to the created instance
2. You should SSH as the '**ubuntu**' user, which contains all necessary information.
3. Change username to ubuntu and connect



4. Enter command in CLI: ./passwd.sh
    1. This script is designed to generate a password for accessing the GUI, such as NICE DCV. When executed, the script sets a password that users can use to log in to the GUI interface of the EC2 instance.



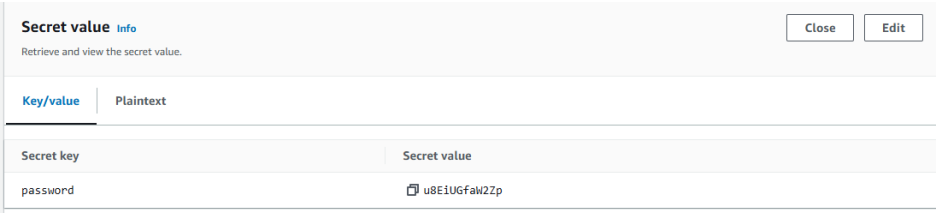
    2. After running the script
        1. Enter AWS Region (default: ap-northeast-1): Enter default is ap-northeast-1
        2. Enter AWS Profile (default: default):
        3. After installing aws cli
        4. Enter AWS Access Key ID (press Enter to keep existing), Enter AWS Secret Access Key (press Enter to keep existing)



As shown in the top image Password updated successfully.
AWS config and credentials updated successfully.

When appears, the password is created normally (when using an image, the ID key must be hidden)

5. AWS Secrets Manager → Secrets → Click on the instance ID → Retrieve secret value of Secret value. You can see that the password has been created.



6. Copy the above password

12 How to Connect via GUI (Graphical User Interface)

Step 1: Navigate to EC2 Dashboard

1. Once logged in, on the AWS Console home page, locate the **Services** menu in the top-left corner.
2. In the search bar, type **EC2** and select **EC2** from the drop-down list.

Step 2: Locate Your EC2 Instance

1. In the EC2 Dashboard, on the left-hand menu, click **Instances** under the **Instances** section.
2. You will be directed to the **Instances** page, where all your running and stopped EC2 instances are listed.

Step 3: Select Your Instance

1. From the list of instances, find the EC2 instance for which you want to check the IPv4 Public DNS.
2. Click on the **Instance ID** of the target instance to view its details.

Step 5: Check the IPv4 Public DNS

1. In the **Description** tab, scroll down to the **Public IPv4 DNS** section.
2. The **IPv4 Public DNS** is displayed here. It looks like this: ec2-198-51-100-1.compute-1.amazonaws.com.

Step 6: Copy the IPv4 Public DNS

1. **Copy** this DNS address by selecting it and using the copy function (Ctrl + C on Windows or Cmd + C on Mac).

Step 7: Access the Application in a Web Browser

1. Open your web browser (Chrome, Firefox, Edge, etc.).
2. In the address bar, paste the copied **IPv4 Public DNS** and append :8443 at the end. The URL should look like this:

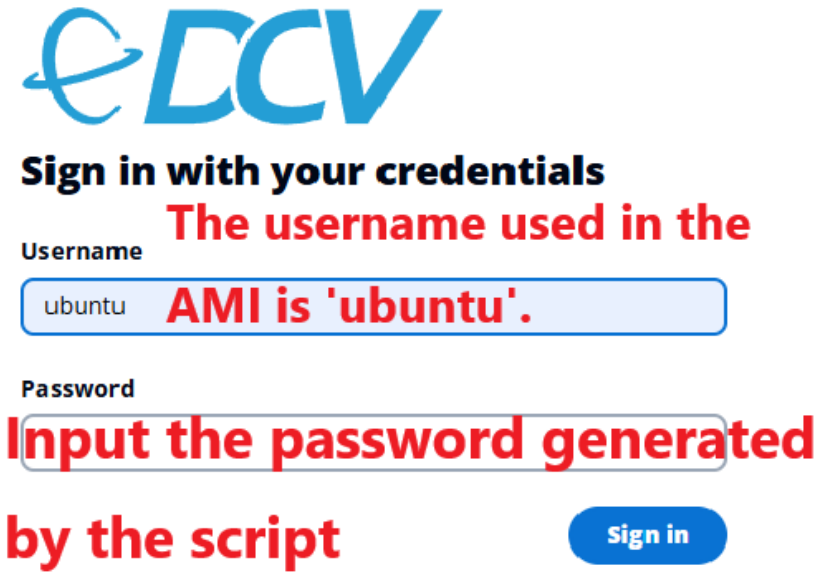
Ex)https://ec2-198-51-100-1.compute-1.amazonaws.com:8443

3. Hit **Enter**.

Step 8: Enter Link

Step 9: Access NiceDcv

1. The username is 'ubuntu', and the password is the one created by the script in Section 10.



Troubleshooting:

- **Cannot Connect?** Ensure that your EC2 instance is running and the correct security group allows traffic on port **8443**.
- **SSL Warnings?** If your application uses a self-signed certificate, your browser might show a warning. You can proceed by clicking "Advanced" and choosing to "Proceed" (not recommended for production environments without a trusted certificate).
- If you encounter a session-related error while logging in (for example, "Session Timed Out" or "Session Unavailable"):

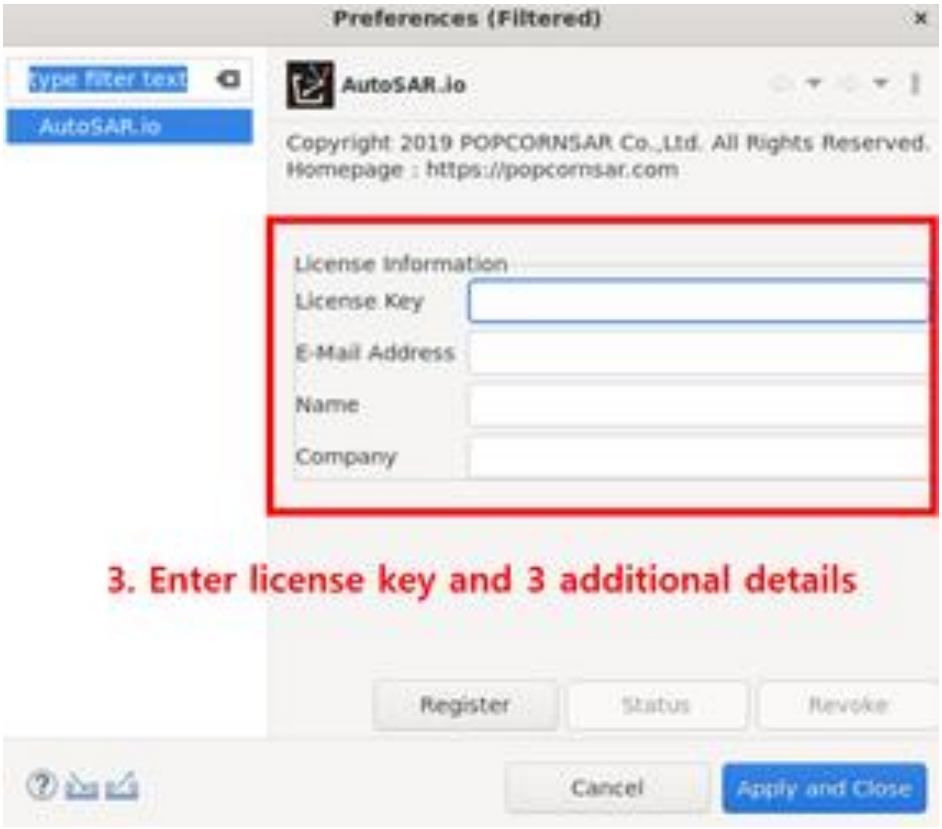
1. Wait for **1 minute** to allow the session to reset.
2. After 1 minute, **refresh the page** by pressing F5 or clicking the refresh button on your browser.
3. Try logging in again using the same credentials

By following these steps, you can successfully access your application on your EC2 instance via the public DNS and port **8443**.

12 Apply a license to your product

Step 1: Click the 'Help' button on the top menu bar

Step 2: Click 'Lincense Key Management'



- Enter the granted license key and enter additional information.

13 Contact

- To contact PopcornSAR support, please email us.
- You can contact us at contact@popcornsar.com or call +82-02-568-3068.
- When updating a license, an updated license is provided.