How to set up remote access for your S-PLAY

Create a convenient, remote access system allowing you or your clients to connect to your S-PLAY from around the world.

Security Note – Internet Connection

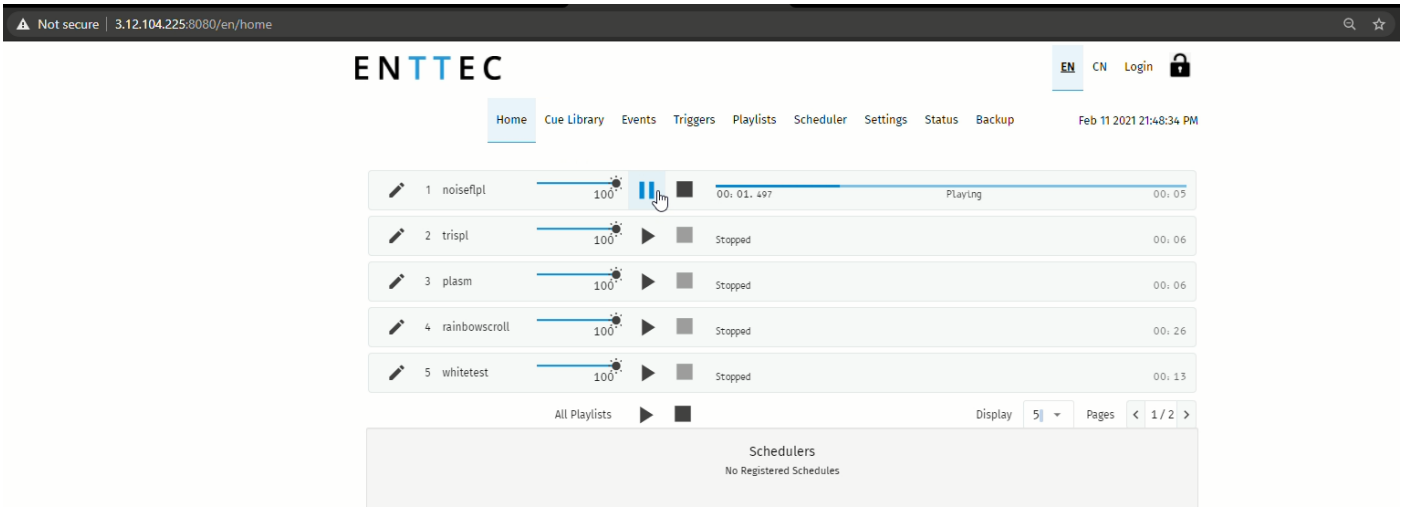
- Before connecting your S-PLAY to the Internet ensure your local network firewall provides security all devices have been adequately secured.
- If ever unsure consult a qualified professional.
- Ensure you have sufficient extra bandwidth to deal with influxes of traffic caused by an internet connection.
- Ensure your SSH Tunnel is configured in such a way to ensure only trusted users can access the tunnel to remotely configure the S-PLAY.

Introduction

In this application note, we are going to learn how to set up a simple remote access system for the S-PLAY alongside using SSH Tunnelling and reviewing other options. The goal is to create a system where we or our customer can connect from a laptop/PC/smartphone from anywhere we have internet access in case we want manual control of our shows.

For the purposes of this guide, we are going to assume that you have already programmed your S-PLAY with your desired cues and playlists – we will be focussing more on the necessary network structure and actual setup.

By the end of this process, we are going to be able to connect to our S-PLAY remotely and see something like this:



1 - S-PLAY - remote access homepage

As you can see this is just the default S-PLAY home page.

What's different is the webpage URL at the top of the screen. We are connecting over the internet through a server with IP address (in this case 3.12.104.225), whilst our S-PLAY is on a completely different address on its local network.

This application note features step by step instructions on how to set up this remote access method with an AWS cloud server. Using these principles, you may choose to set up your server with a different service using this application note for reference.

Getting started

Requirements.

Before you begin, we will need to have the following:

1. S-PLAY with Internet access – this can be through a 4G router or just by being connected to a network with internet access.
2. AWS account – you can sign up for a free account with AWS here: *https://aws.amazon.com/*
3. A computer connected to the S-PLAY and internet so you can set up the remote access function.
4. (Optional) a smartphone or other internet connected device that you can use to test the remote access function once it is set up.

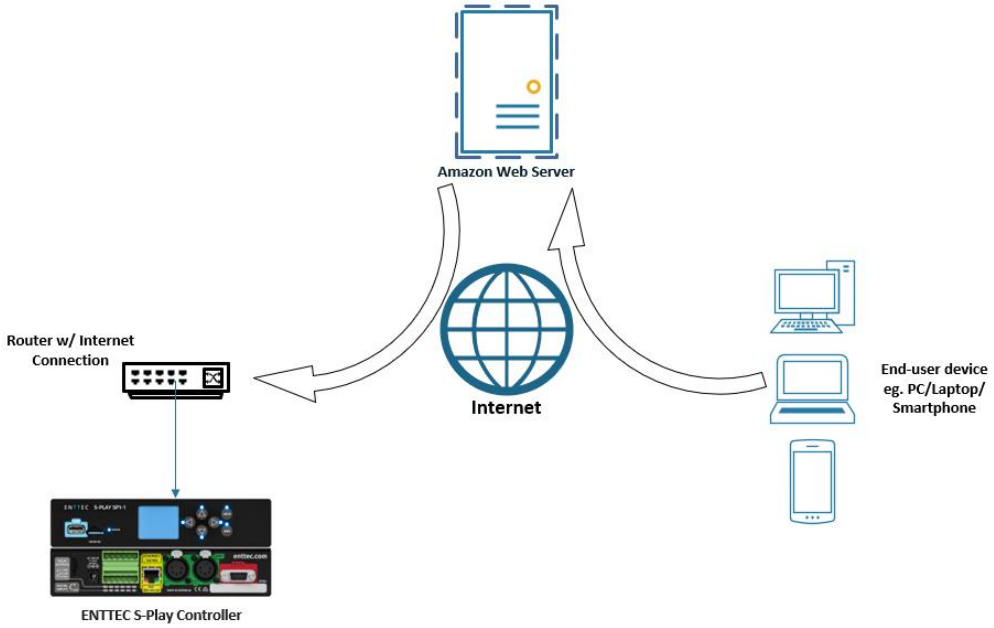
Remote connection options + network structure

Simple remote connection options.

If you have worked with other network-based devices and control systems before you are probably already familiar with some other remote access methods including:

- Remote Desktop Access through a local computer – using a service like TeamViewer, LogMeIn, or RealVNC.
- Setting up port forwarding on a 4G router.

This remote access function, however, is designed to streamline the process for the end-user so they can use a simple URL and connect from anywhere, providing they have secure internet access. A high-level diagram of this application can be seen below:



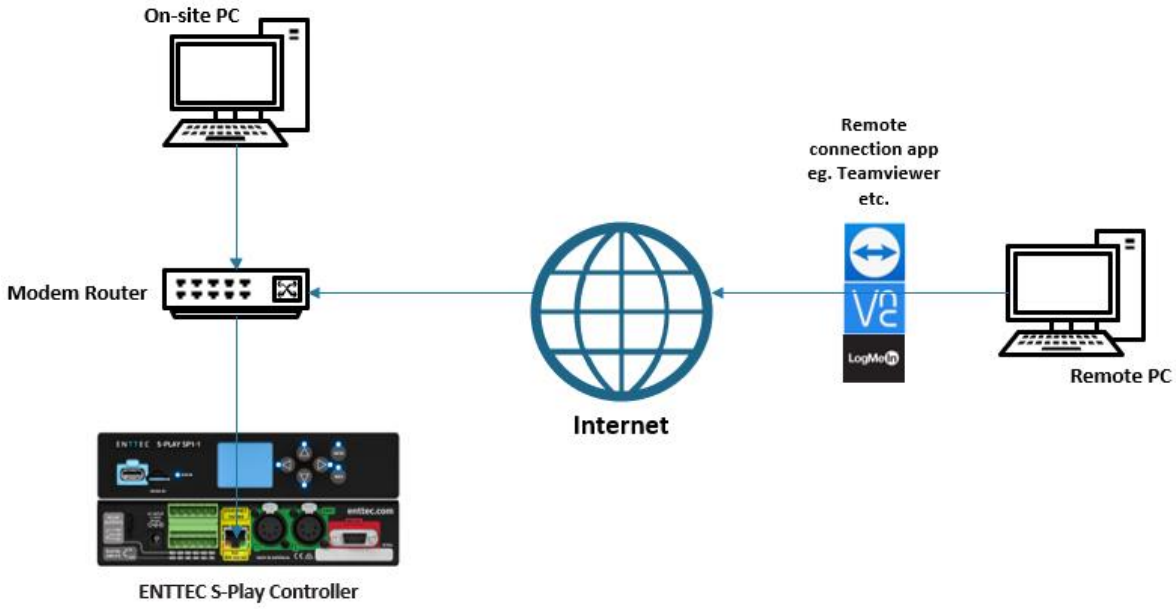
2 - Remote access network structure

As you can see, this method uses an AWS cloud server as the intermediary to connect your device to the S-PLAY. This seems a bit counter-intuitive… why use a separate web server to make this connection? Why can't we just connect over the internet straight to the S-PLAY?

Well, we can, using the 2 methods listed before. Let's take a look at those:

Remote Desktop Apps

The most straight forward way to connect remotely is to use an app like TeamViewer or RealVNC to give access to a PC on the same network as the S-PLAY that you can treat as if you are on the same network:



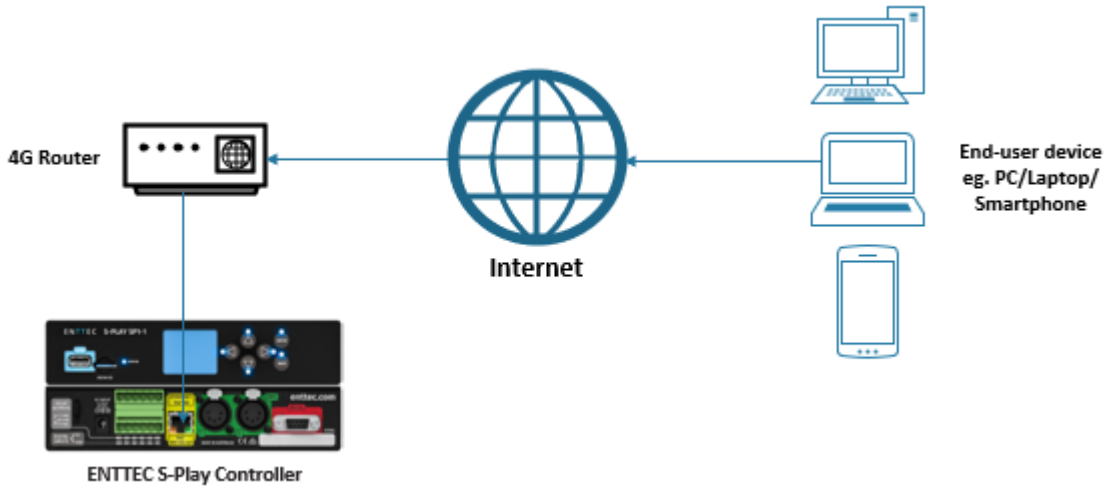
3 - Remote desktop network structure

As you can see, this method requires that we have a PC on-site that is connected to our lighting control network. The remote desktop apps will connect us to the PC, and then using that remote connection, we then navigate through the on-site PC to connect to our S-PLAY as if we were there in person.

The limitation of this method is that we need to have a PC on site, and it needs to be on all the time if we are to connect to it remotely at any time.

Port forwarding on 4G routers

How about 4G routers? You can get 4G routers inexpensively and add a data SIM card to them to allow remote internet connections. You can also then set up a port forwarding rule so that whenever you connect to that router, you are re-directed to the S-PLAY



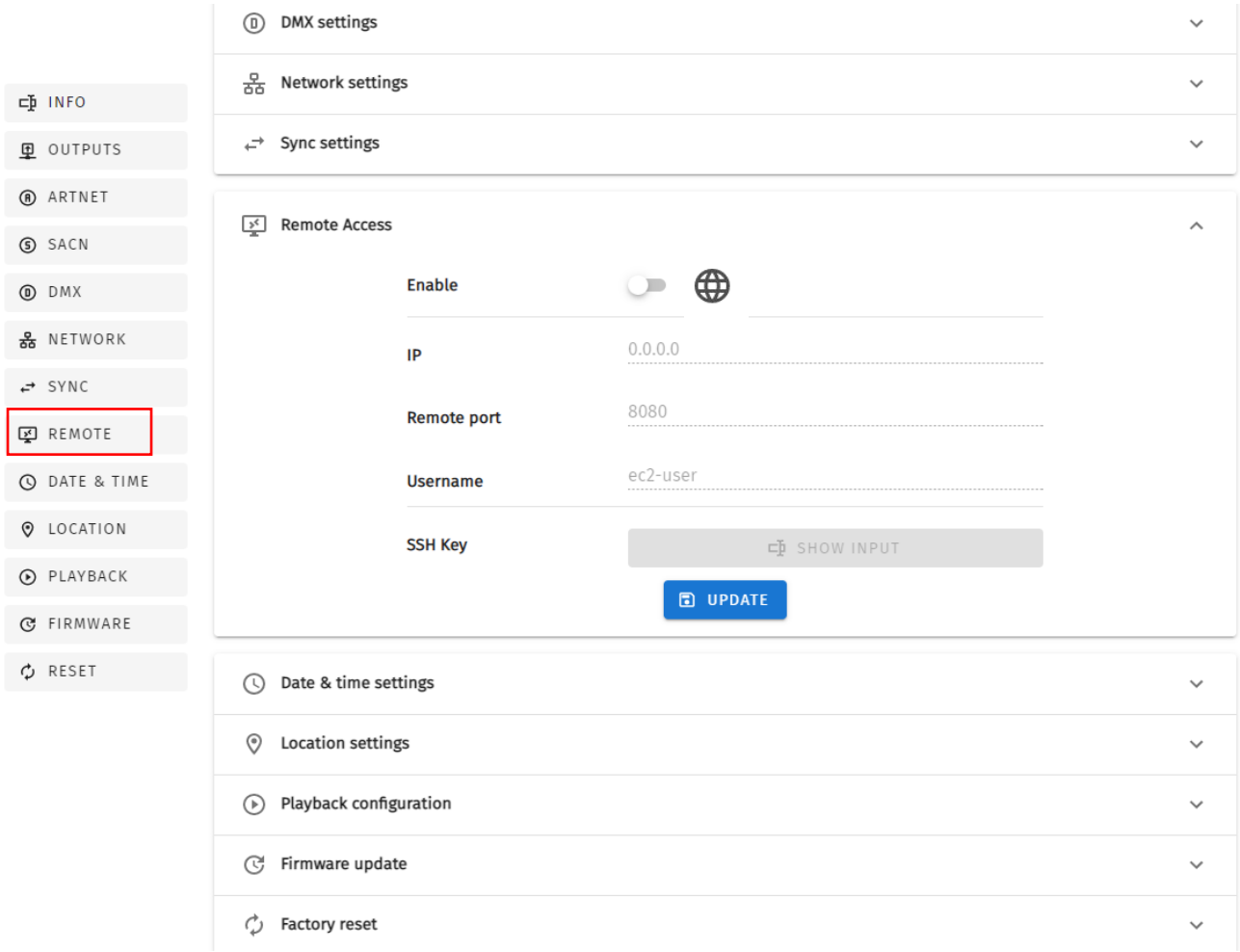
4 - Port forwarding network structure

The limitation with this method is that you need a public IPv4 address on your 4G router. Depending on where you are and what your ISPs can provide, this might be a bit difficult. For example, here in Australia, most mobile services use CGNAT which means your public ip can change many times in a hour, so you may have to get specific, and much more expensive business internet plans to get a fixed IP.

This brings us to the method using the S-PLAYs new remote connection feature. Using a separate web server to help with the connection eliminates the need for an expensive business-level data plan like we needed for port forwarding. Instead, we'll set up a cloud server once, then the S-PLAY will give us a URL that we can use to connect to it from anywhere in the world as long as we and the S-PLAY both have internet access. Stay tuned as we'll be running through how to set up this cloud server later on.

Setting up the S-PLAY

Firstly, let's update our S-PLAY and see how this all works. In Settings, select Remote in the menu, we'll see a section titled: "Remote Access" It's asking for an IP Address, port number, username and SSH Key. These are all obtained when we set up our virtual server.

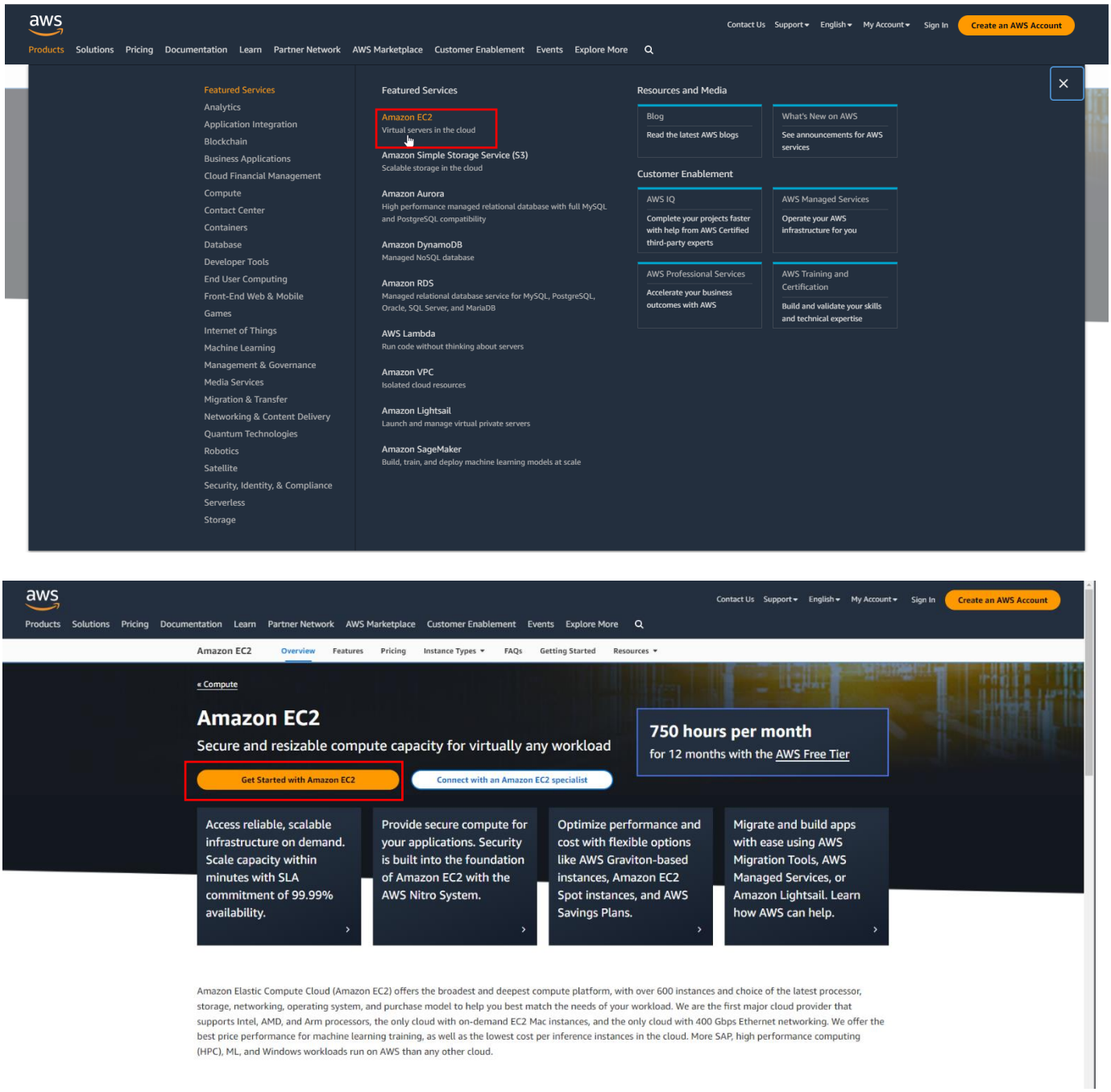


5 - S-PLAY remote access settings

We'll come back to this later once we have created our AWS server.

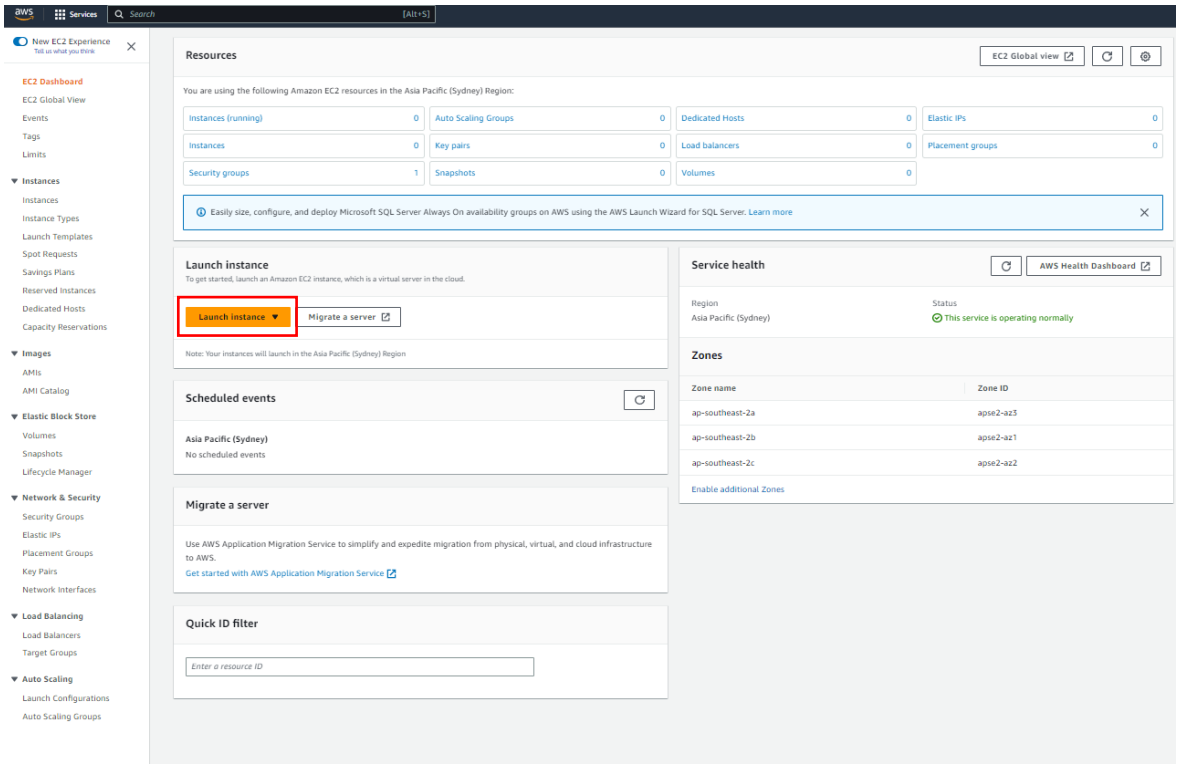
Setting up the cloud server

1. We'll be using Amazon Web Services as an example on how to set up a simple cloud server for remote connection. We won't go through how to make an account – that's straight-forward, but once you've created a free AWS account, you'll want to log in and look through the various services being offered. We need a "Compute" service for this function, and we'll use the EC2 version since it's free tier eligible and has the functionality we need.



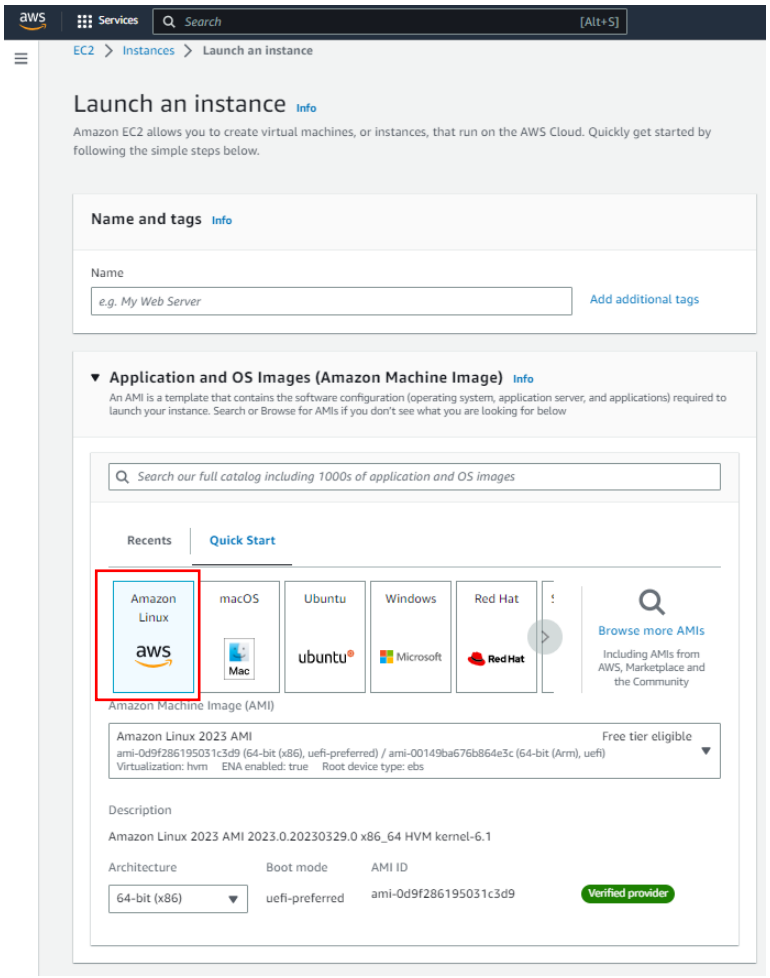
6 - AWS Server type selection

2. Next, we'll go to "Launch Instance" to create a new server instance.



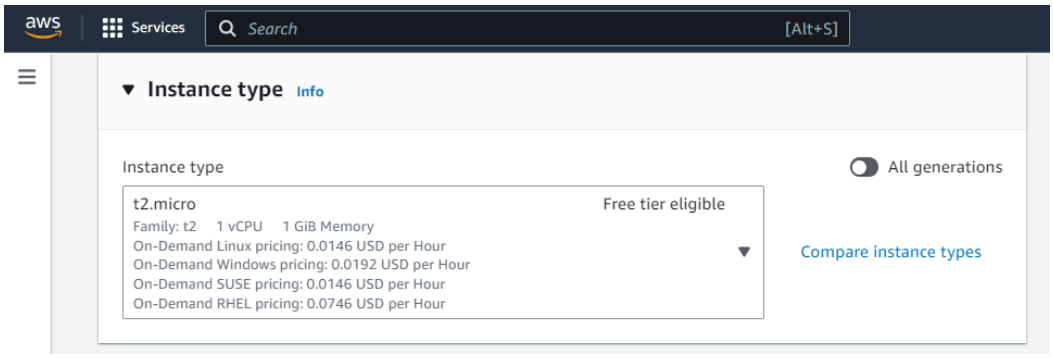
7 - AWS – Launch Instance

3. Create a name for this instance and for the Application and OS Images, we'll just chose Amazon Linux, again because it's free tier.



8 - AWS – Application and OS Image Selection

4. For the Instance type we'll just go with t2 micro because – you guessed it - it's free and we really don't need this to be too powerful for what we need to do.

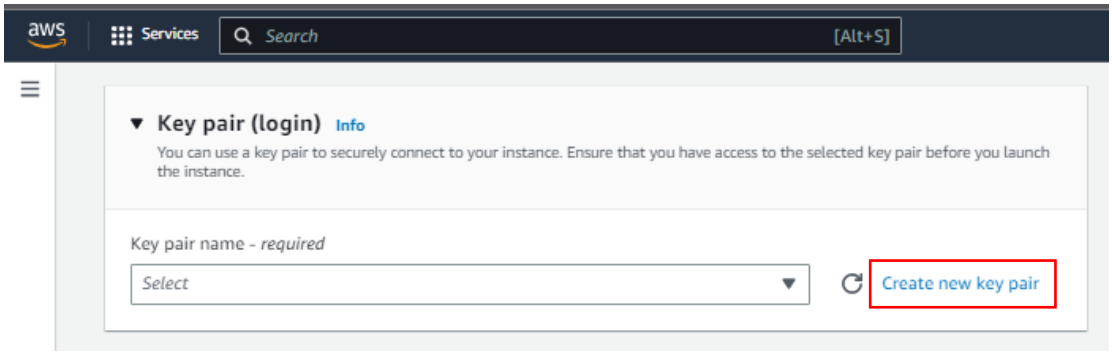


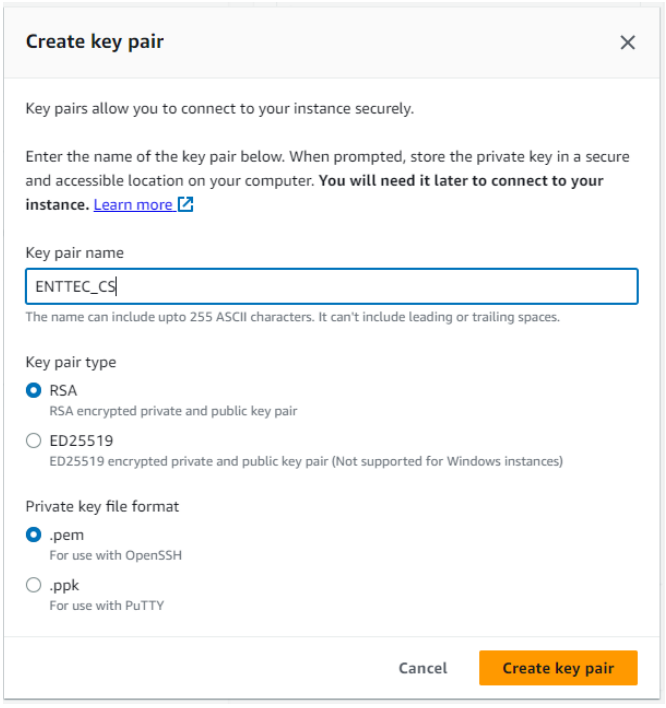
9 - AWS – Instance Type Selection

5. Key pair is a key file that your S-PLAY will need to be able to connect to the AWS server. Create a new key pair and save this in a safe location. The key pair will be in the form of a .pem file which you will need to upload to your S-PLAY in the SSH Key field at the end of the settings.

Note:

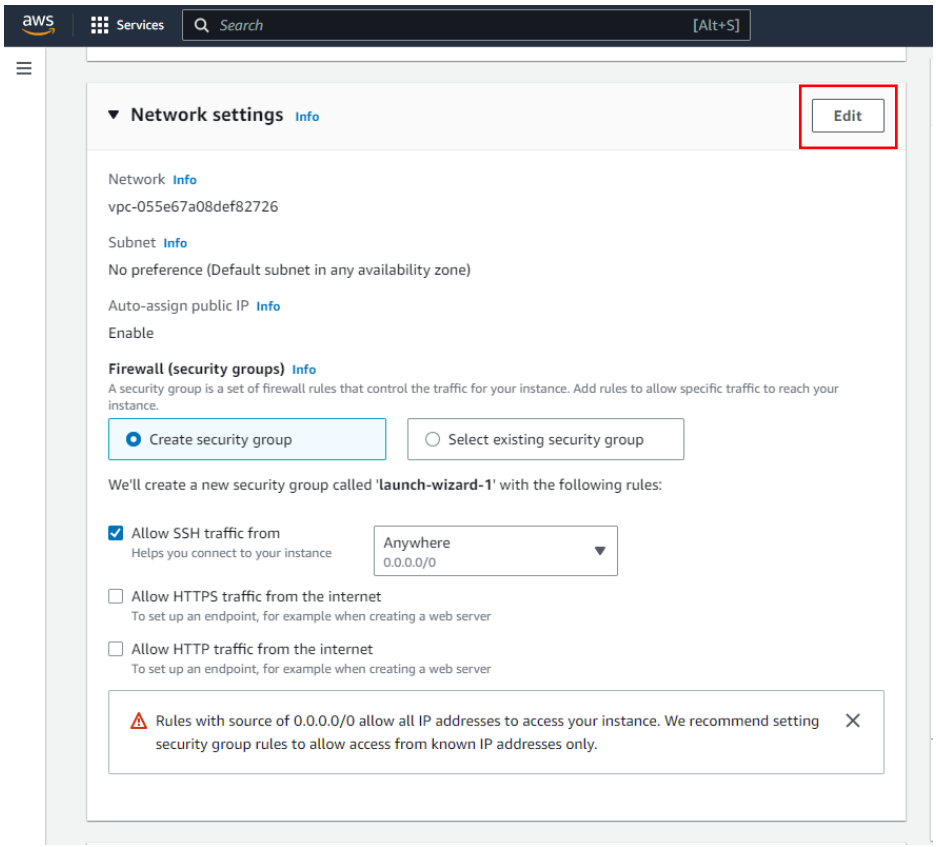
- Ensure to keep this file as there is no any way to access it again if the file is lost.
- Disabling remote access on the S-PLAY will wipe all SSH configuration including the .pem SSH Key from the S-PLAY for security reasons.





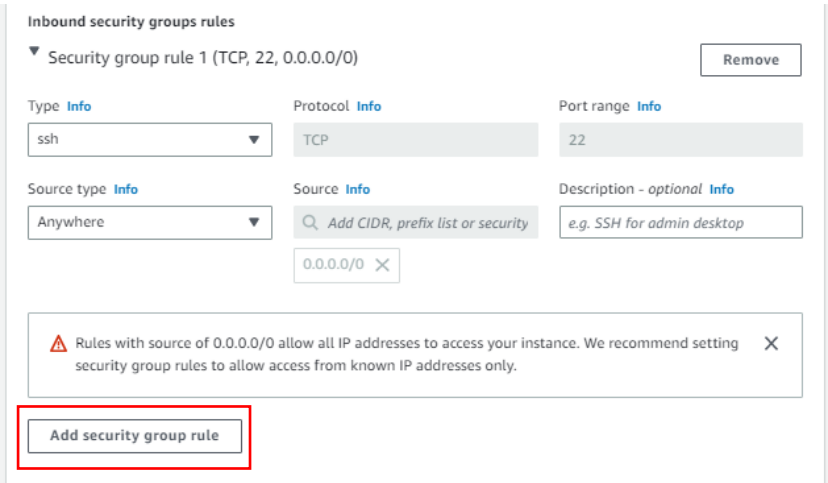
10 - AWS - key pair creation

6. Network Settings is where we add our port settings so that any device can access the server. Click 'Edit' on the right to access more settings.



11 - AWS – Network Settings

7. By default, the SSH rule (Port range 22) will be present in Inbound security groups rules, we just need to change the source to "anywhere". In addition to this we need to add exceptions for 4 additional port ranges: 80, 8080, 13133 & 55555 by clicking 'Add security group rule'. These are ports that the S-PLAY will be using to display its web page and allow interactivity to the connecting device.

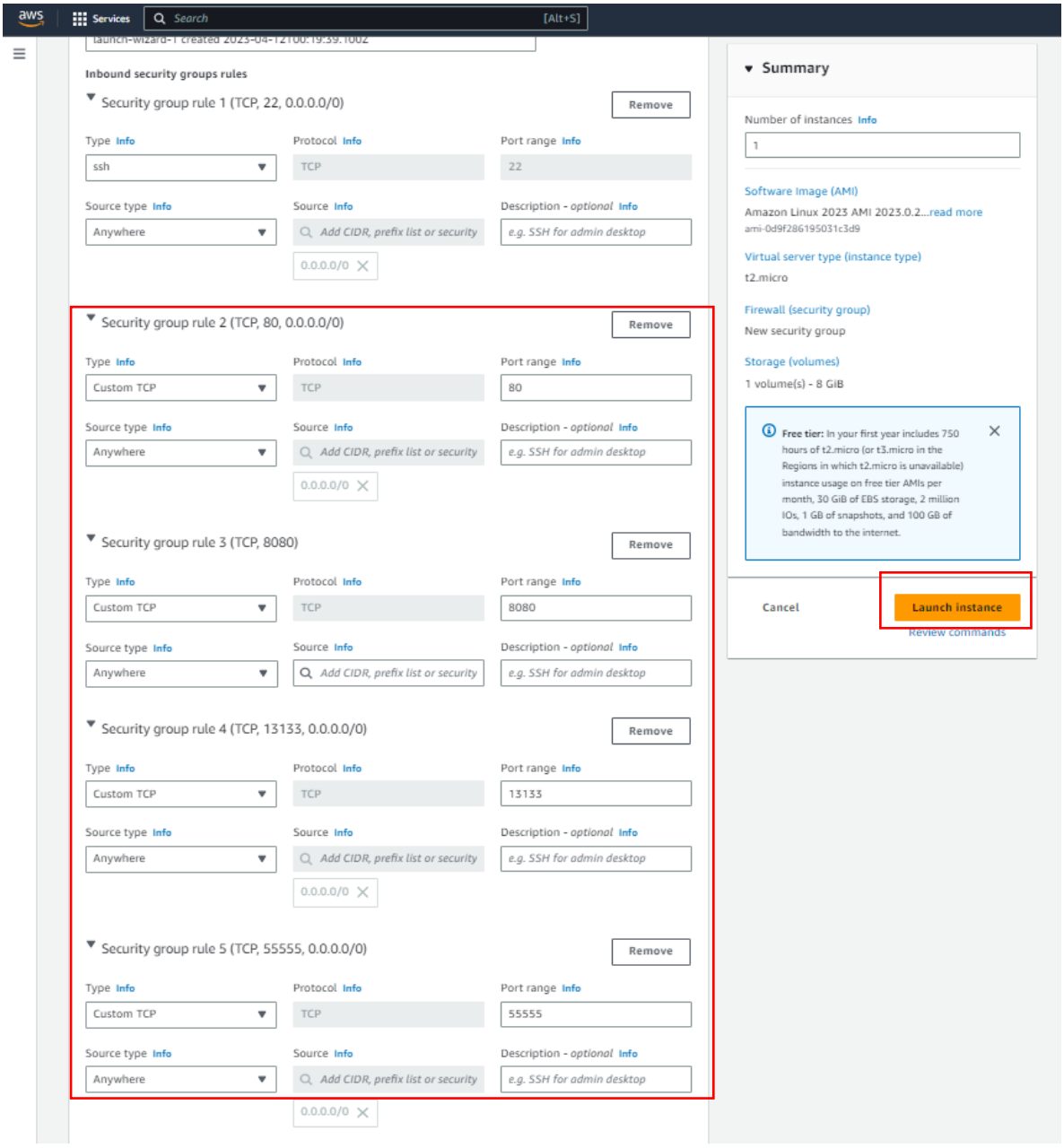


12 - AWS – Add Inbound Security Group Rules

8. Enter Port Range and change Source Type to 'Anywhere' for each 4 port ranges so that any device can connect to our S-PLAY no matter where they are.

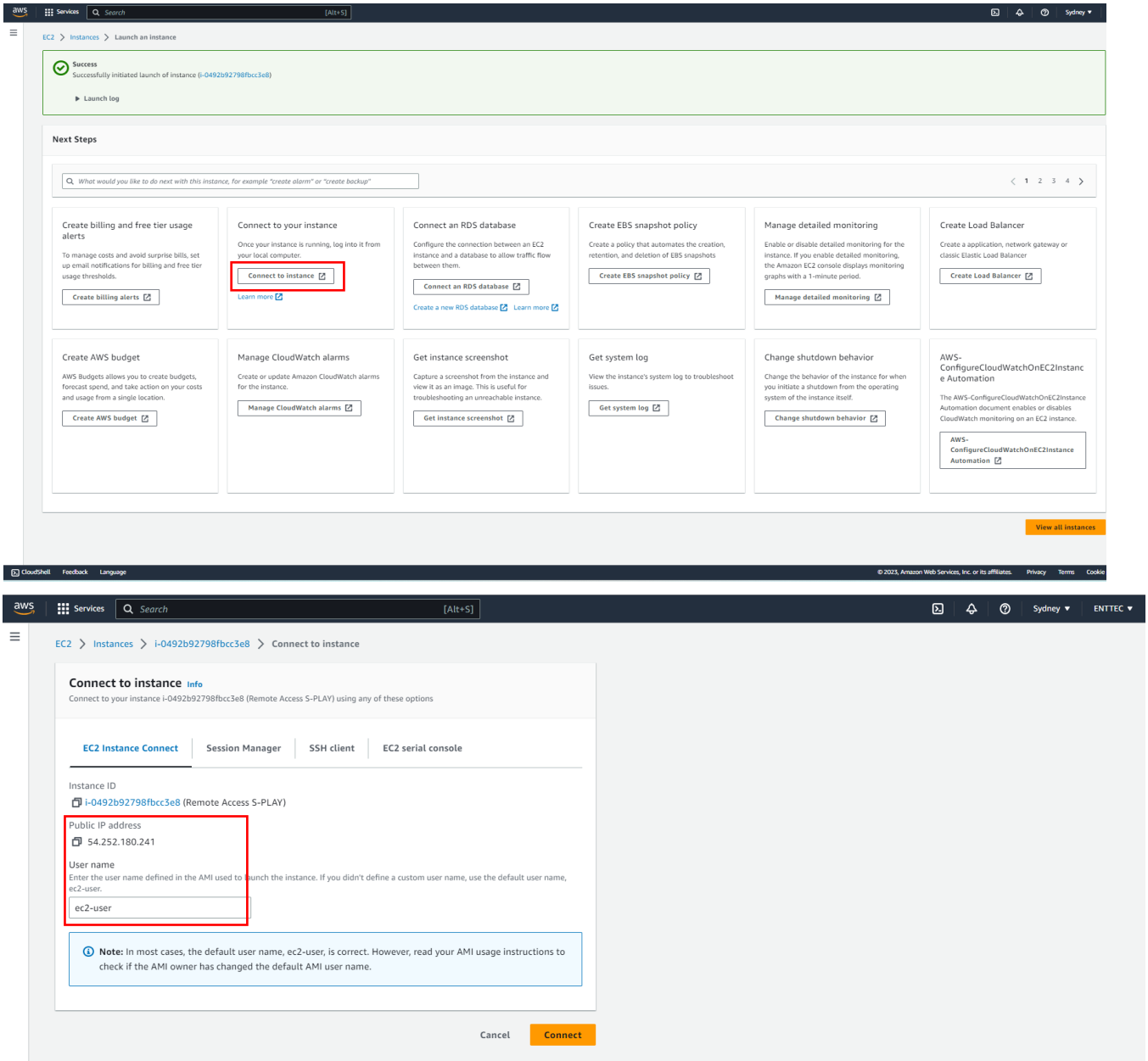
   You can improve the security of the system by limiting the source ranges. For example, if you work for an integration business and you are installing S-PLAYs on your clients' projects. You can set the source to be the client's office IP address range. This means only a device in the client's office can access the S-PLAY. You could also set this to your own office IP address range if you were to make changes/updates on your clients' behalf.

   Once the rules in the previous section have been set, the new server instance is ready to be launched by clicking the 'Launch instance'.



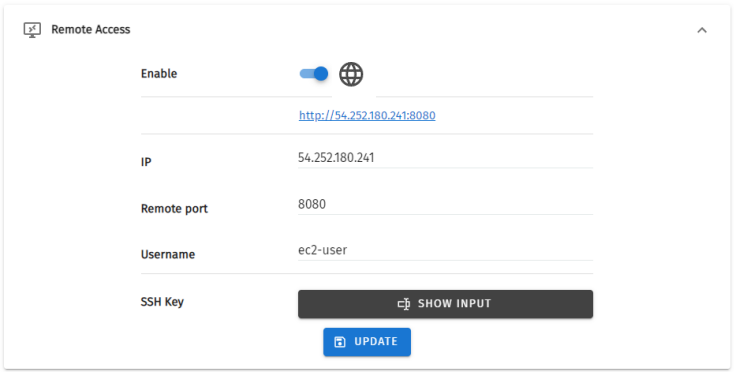
13 - AWS - Port Settings & Launch Instance

9. After the instance is successfully created, it will bring us to the “Next Steps” menu. From here we click “Connect to instance” which brings us to this screen showing us the public IP address and Username of our instance.

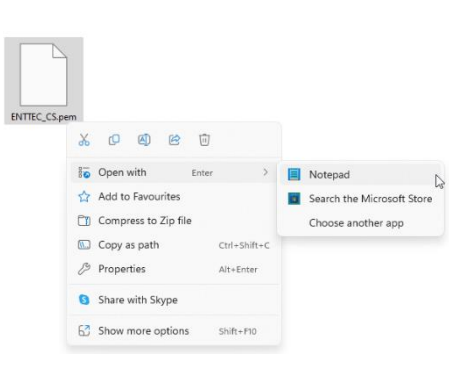


14- AWS - connect to instance

10. We can enter both these values into the appropriate lines on our S-PLAY remote access settings. Then, open the .pem file created from Step 5 with Notepad and copy the key into the SSH Key field. Click update to ensure those are saved.



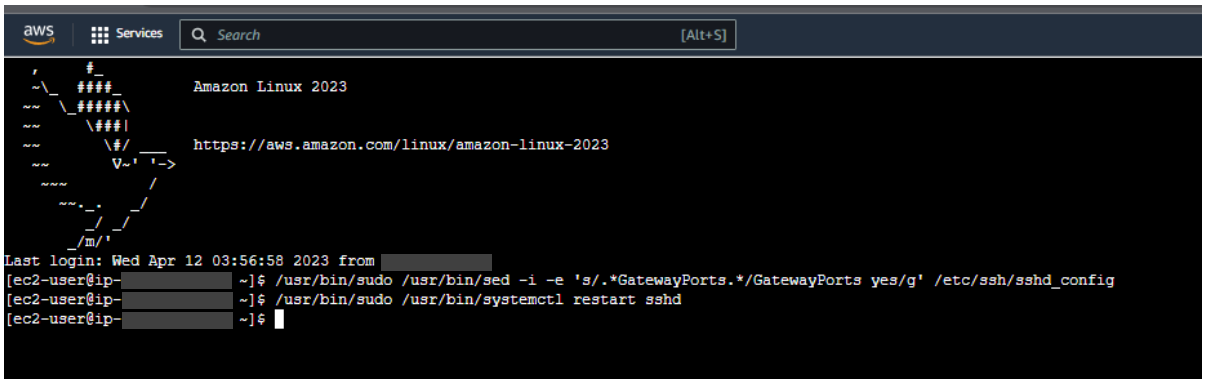
15- S-PLAY - update remote access settings



93 – PC – Key pair .pem file to be entered in SSH Key section

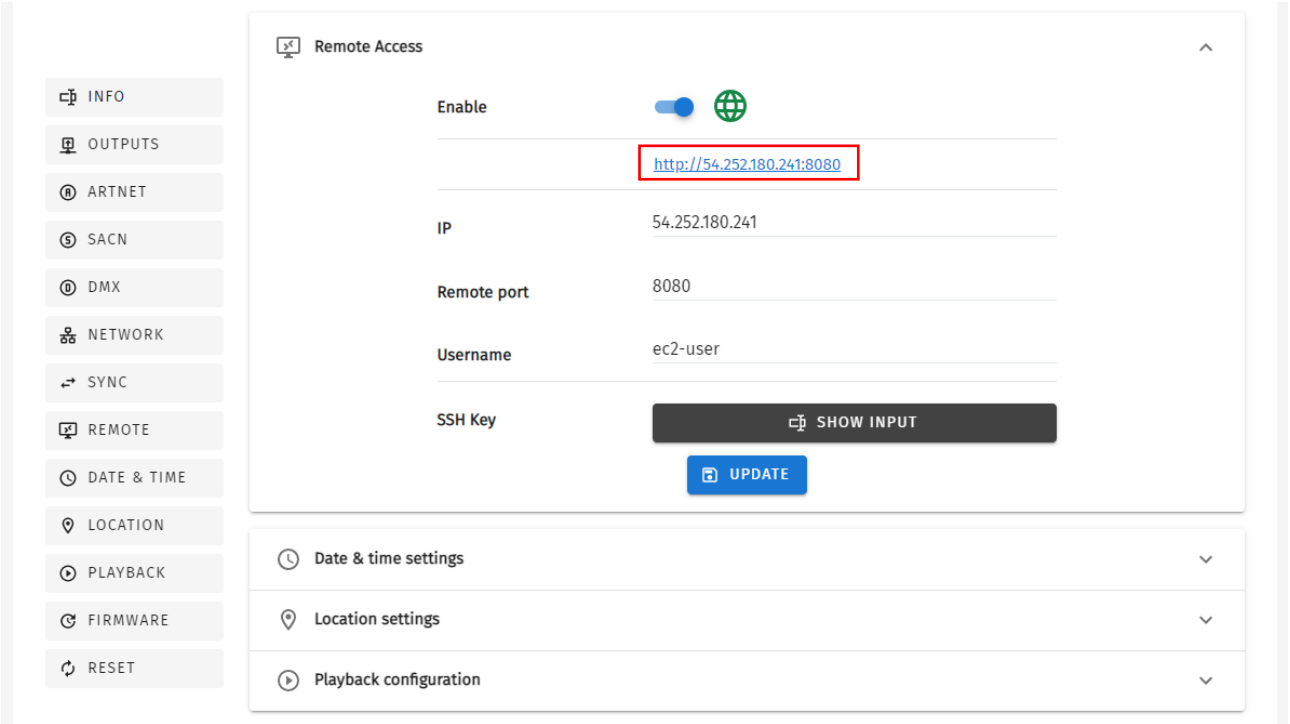
11. The last step in this configuration process is to modify an access setting on our newly generated AWS cloud server, so the S-PLAY can access it. To do this, connect through to your instance. This opens up a new tab and a command prompt window. To update the setting, we've put together a short code segment to go through and make the necessary changes.

    Here is the code segment you will require:
    /usr/bin/sudo /usr/bin/sed -i -e 's/.*GatewayPorts.*/GatewayPorts yes/g' /etc/ssh/sshd_config
    /usr/bin/sudo /usr/bin/systemctl restart sshd



16 - AWS - server command prompt window

12. After copying that code segment in, the necessary update is made. We can now go back to our S-PLAY screen and use the given URL to connect remotely. You should notice that the globe icon has now changed to green – indicating that the remote access setting is active and the URL will allow S-PLAY access from your smart phone or any other device from other network.

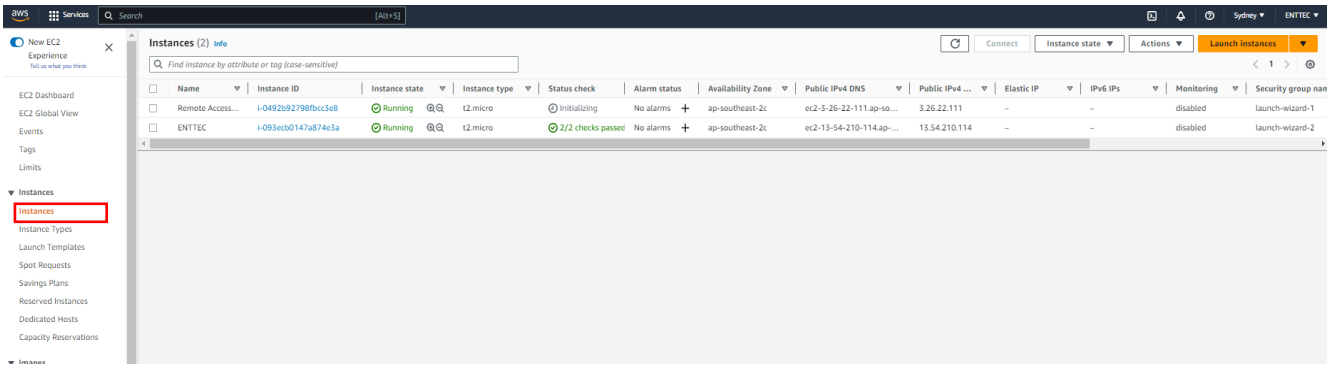


17 - S-PLAY - successfully updated remote connection settings.

Remote access multiple S-PLAYs in one server

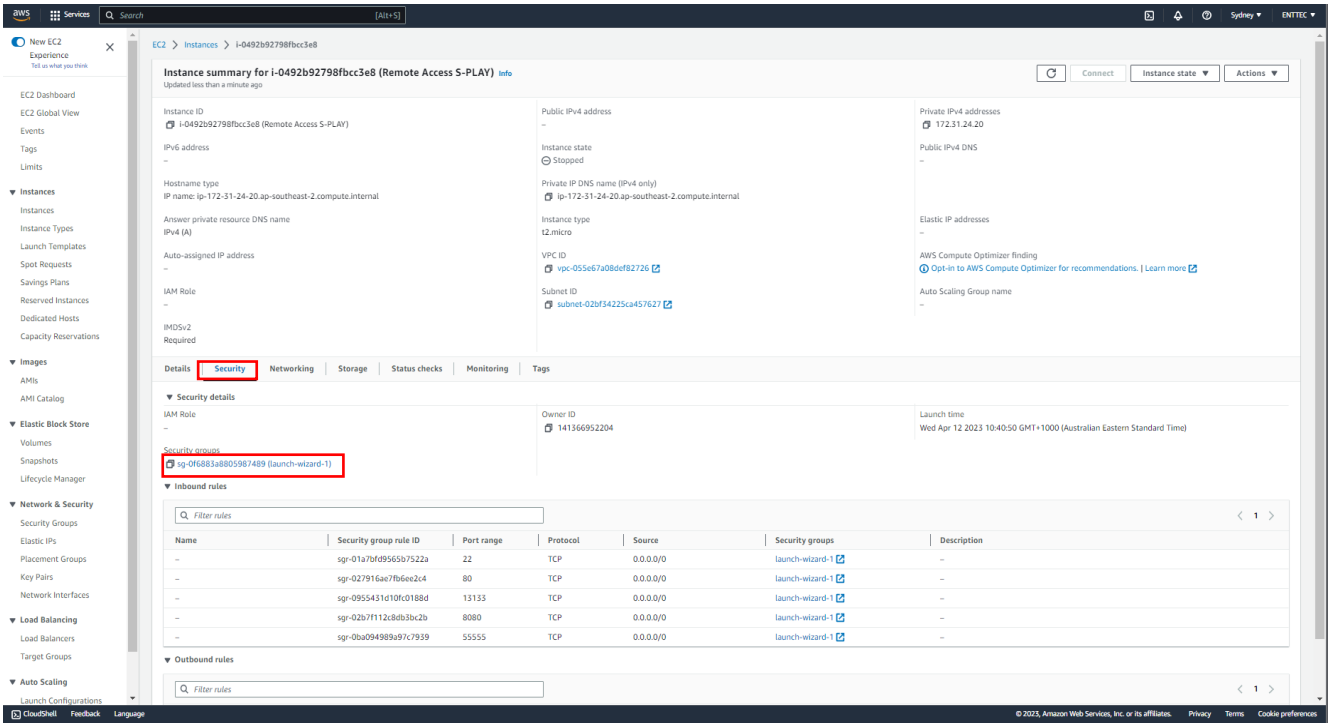
If you have multiple S-PLAYs, they are allowed to be accessed remotely within the same AWS cloud server. Simply add extra ports in the exiting cloud server by the following guide and configure multiple S-PLAYs to be accessible via one single AWS.

1. Click 'Instance' from the menu on the left, select the existing 'Instance ID' from the list for the additional S-PLAY to be added in. This brings us to the instance summary page showing the instance settings.



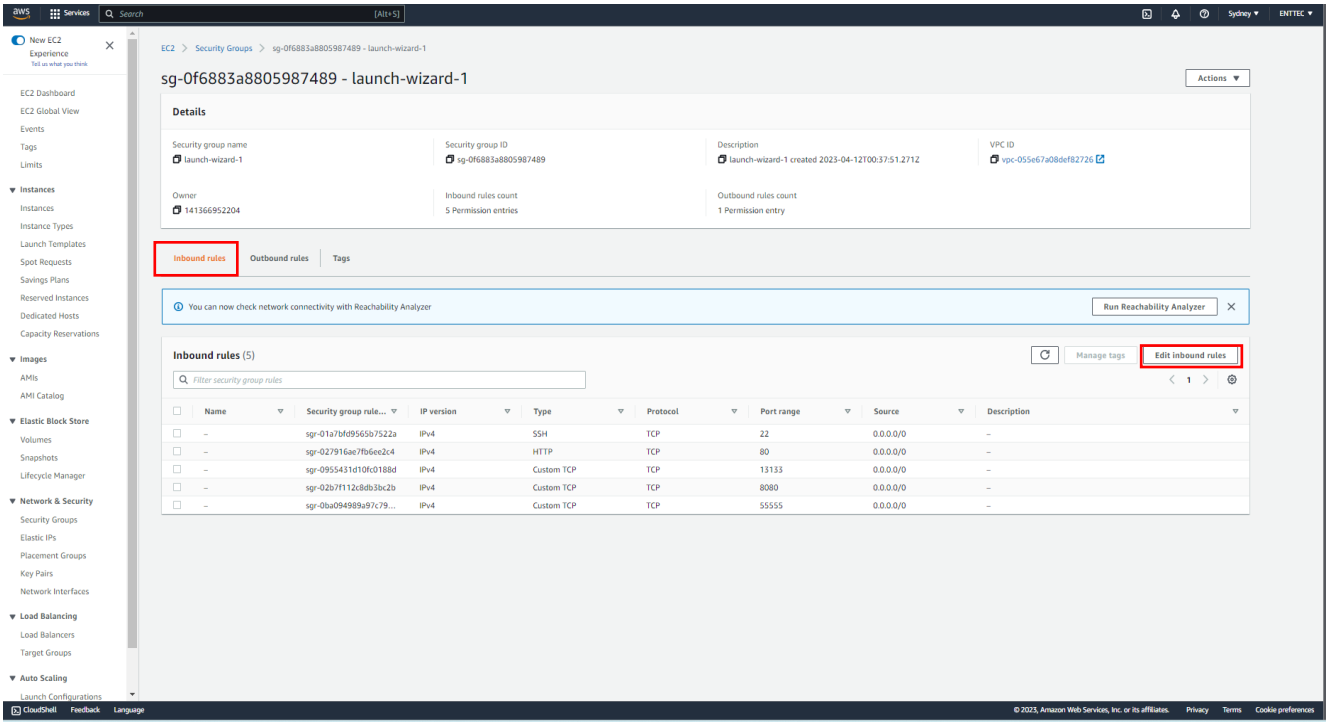
18 – AWS - Add extra S-PLAY into the existing server from AWS EC2 Menu.

2. In Instance summary page, navigate to 'Security' and select the 'Security Groups'. From here it will open up a page to allow Inbound rules to be edited.



19 – AWS - Instances configuration summary page.

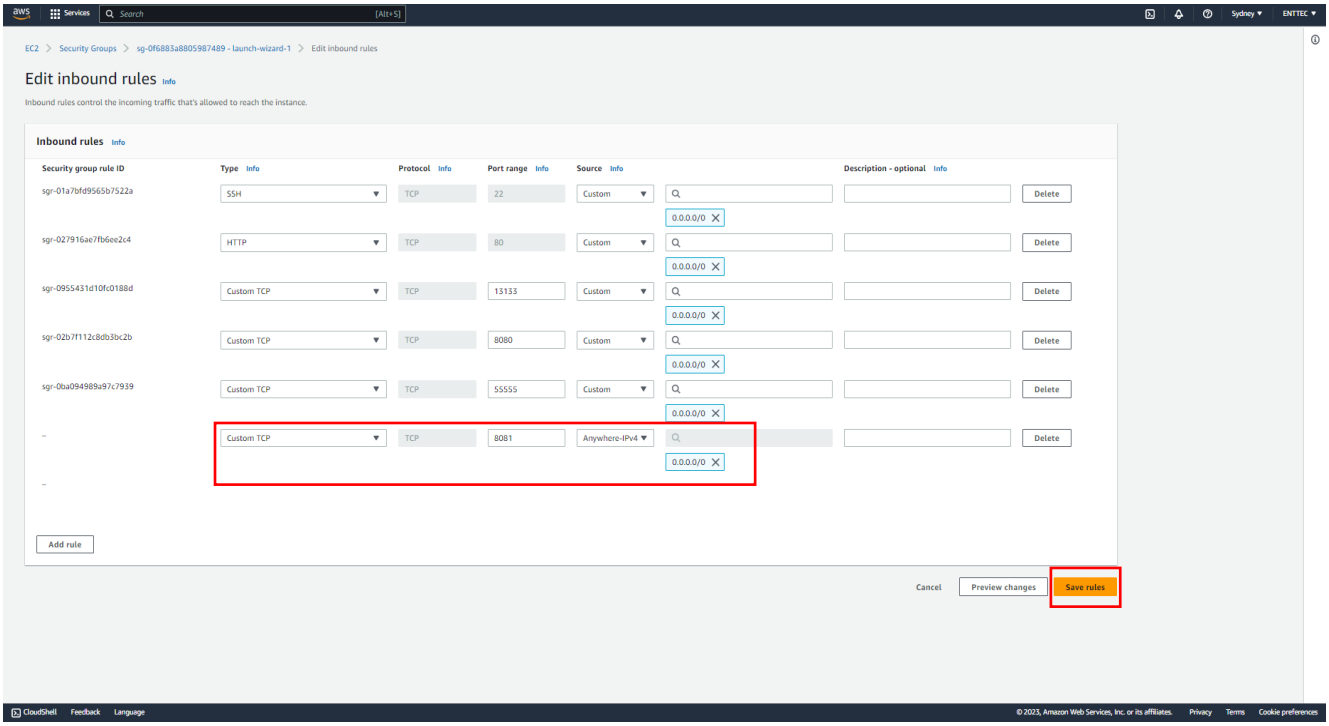
3. Select ‘Inbound rules’ tag and click on ‘Edit inbound rules’.



20 – AWS – Security Group Settings Page.

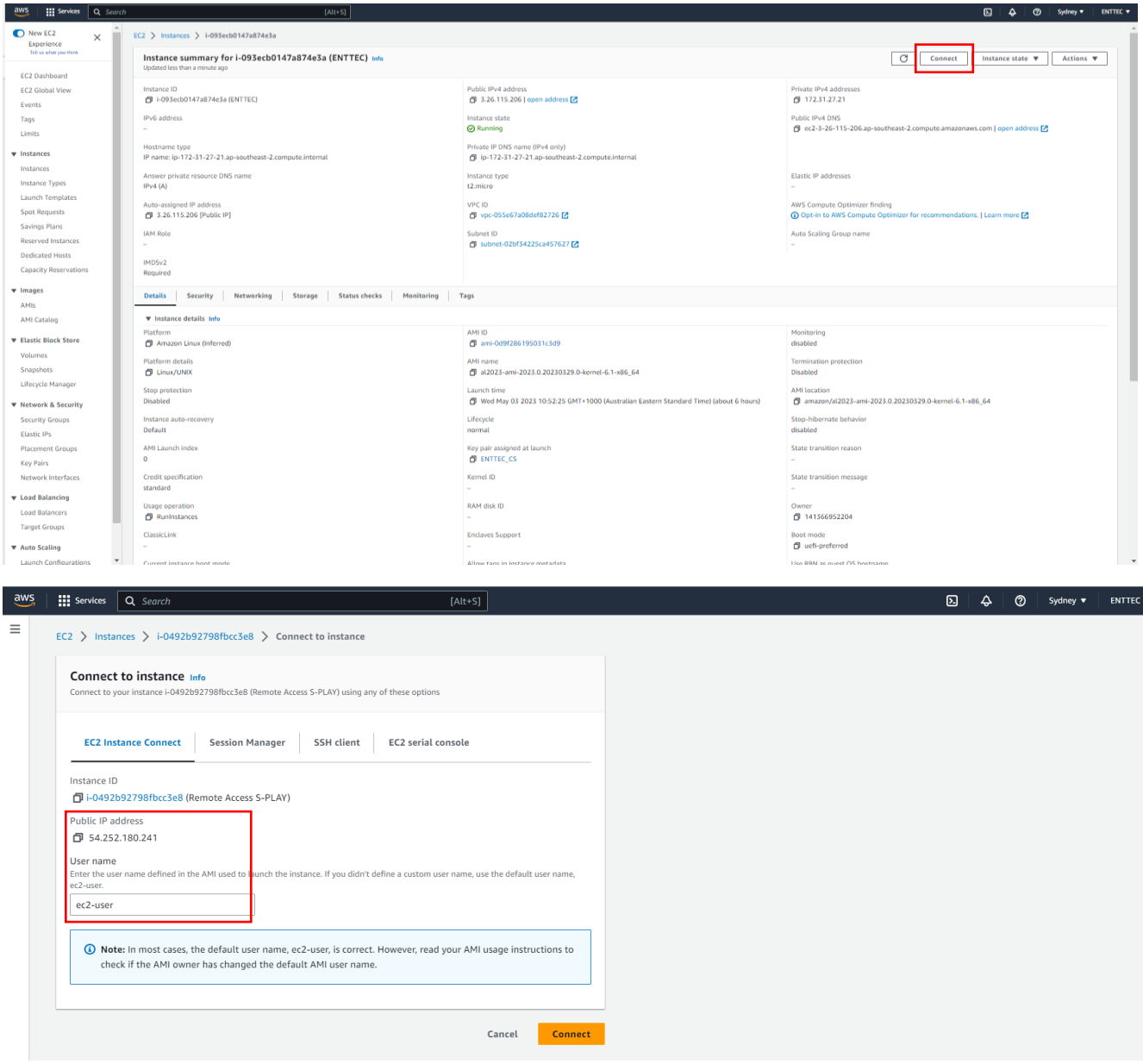
4. Click ‘Add rule” and set ‘Port range’ to be a value between 1000 to 65000 (excepting 5555) and ‘Source’ to be same as above ports ‘0.0.0.0/0’. You can create multiple ports here all at once to allow all your S-PLAYs to be accessed from this server. Don’t forget to ‘Save rules’.

   Here in the below example, one extra port 8081 is added for the additional S-PLAY and we will use this when setting up the S-PLAY.



21 – AWS – Editing inbound rules to add extra port.

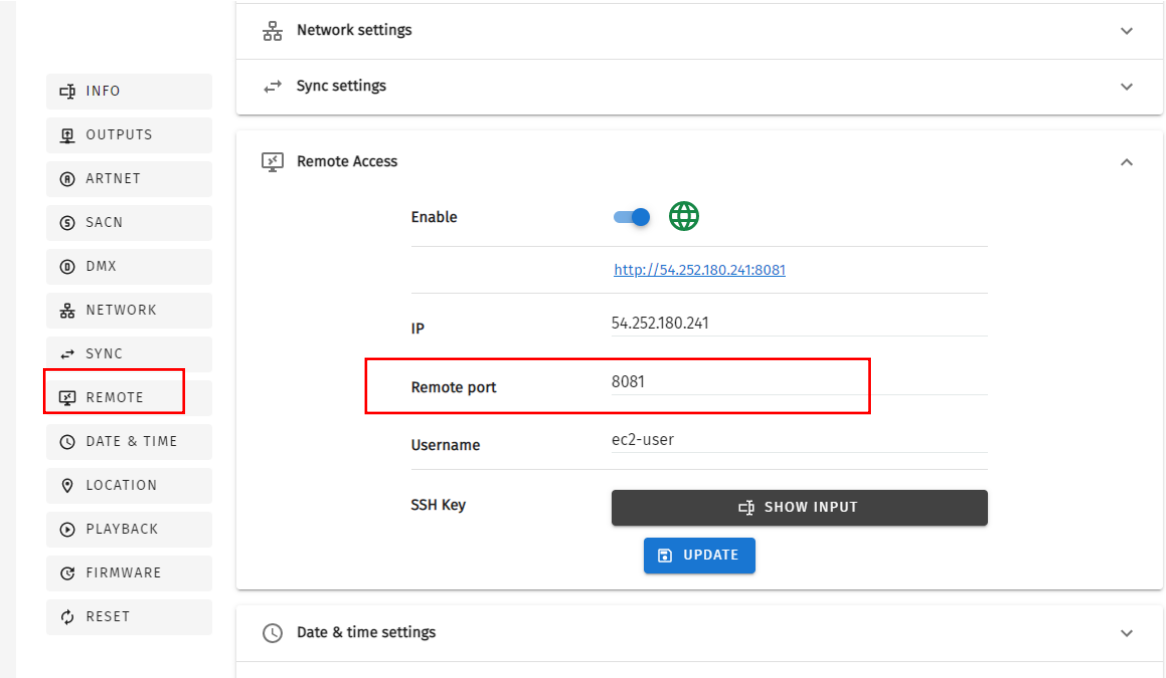
5. Before we configure our S-PLAY, the IP address and the Username from the server are required. Go back to the instance summary and click 'Connect' button on the top right corner that will navigate us to the page showing the required information.



22 – AWS – Retrieving IP address and Username of the cloud server.

6. Now we can start setting on the additional S-PLAY. Access to the S-PLAY webpage and navigate to Settings tab. Enable the Remote Access from your additional S-PLAY, enter the IP, Username and SSH Key (the .pem Key pair file created in Page 8) while the Remote Port at the value set from the last step.

   Here in our example is 8081. Press Update, and the remote access globe sign should turn green which means it is active.



23 – S-PLAY – setting up the newly created port to the additional S-PLAY.

Remote Access: Troubleshooting

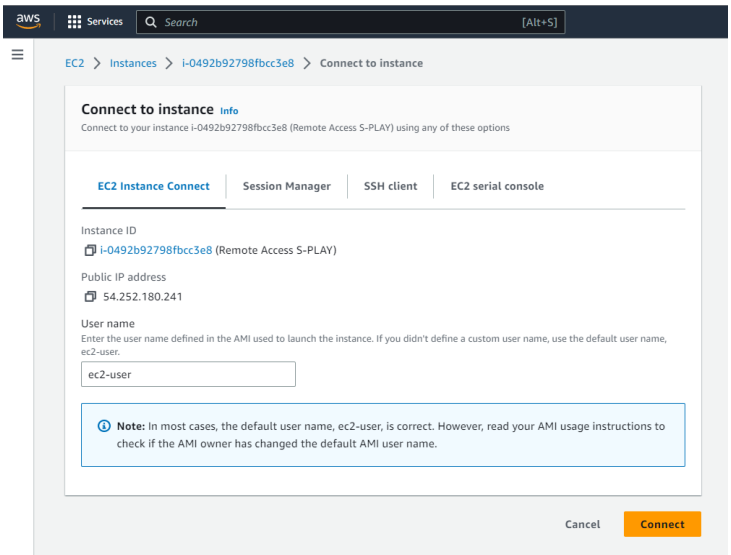
On first connection, the S-PLAY will automatically try to configure the server's gateway by running:

```
/usr/bin/sudo /usr/bin/sed -i -e 's/.*GatewayPorts.*/GatewayPorts yes/g' /etc/ssh/sshd_config

/usr/bin/sudo /usr/bin/systemctl restart sshd
```

If the Username provided in setup doesn't have any sudo rights or sshd_config is located in different place, you will need to manually update the sshd_config on your AWS Server to set "GatewayPorts yes" and restart the sshd service.

To access the terminal, navigate to the server instances on AWS. Press Instance ID and then 'connect' on the top right corner which brings us to the page as below. Click 'Connect' again to open the terminal.



To allow the S-PLAY to manage your AWS server's gateway in needs to permit this, run:

```
/usr/bin/sudo /usr/bin/sed -i -e 's/.*GatewayPorts.*/GatewayPorts yes/g' /etc/ssh/sshd_config
/usr/bin/sudo /usr/bin/systemctl restart sshd
```

It's worth noting that. /etc/ssh/sshd_config can be located in a different folder depending on the operating system running on your remote server.

Wrapping up

That brings us to the end of the configurations we need to do for remote connections but remember that we set up this server to be accessible from any IP address, so for security it's highly recommended to use the S-PLAY's password lock function. You can do that by going to your S-PLAY's home page and clicking "user" drop down and clicking "change passwords".

To connect remotely, just go back to our remote access settings and look for the URL that the S-PLAY displays. This address is generated by the S-PLAY according to the settings you have just input and is the address you need to enter to access your S-PLAY remotely.

Conclusion

By following these steps, you can create a control panel on your smart phone or tablet to intuitively control the S-PLAY, call shows, and adjust brightness's. This guide gives you a run down on a basic workflow and control panel, but this is just the beginning. With a bit of time and ingenuity, you can create even more sophisticated control panels.

This brings us to the end of this guide on how to set up remote access on your S-PLAY. By following these steps, you can set up your S-PLAY to be accessible from around the world over the internet.

We hope you found this application note useful!