

# 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



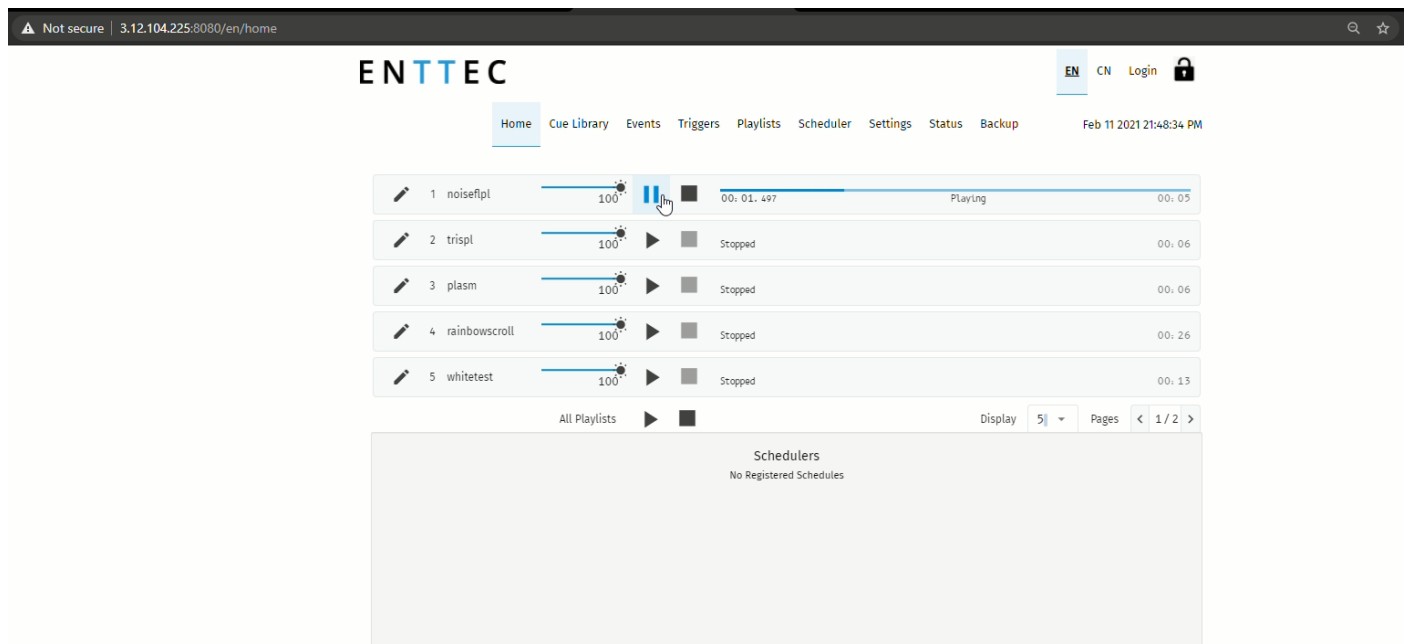
- Before connecting your S-Play to the Internet ensure your local network firewall provides security all devices have been adequately secured.
- If ever unsure consult a qualified professional.
- Ensure you have sufficient extra bandwidth to deal with influxes of traffic caused by an internet connection.
- 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.

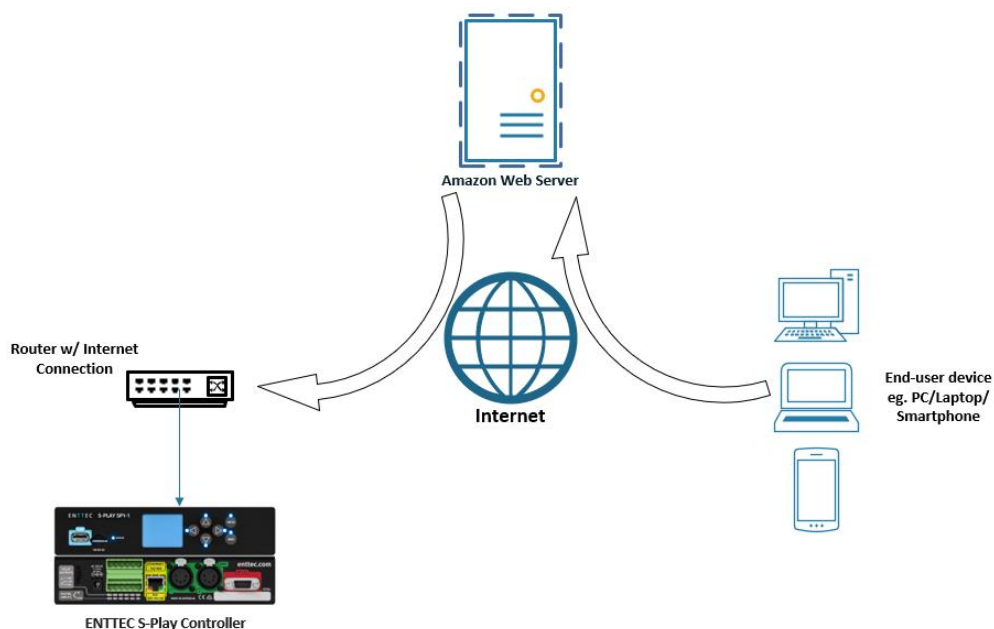
## Remote connection options + network structure

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:

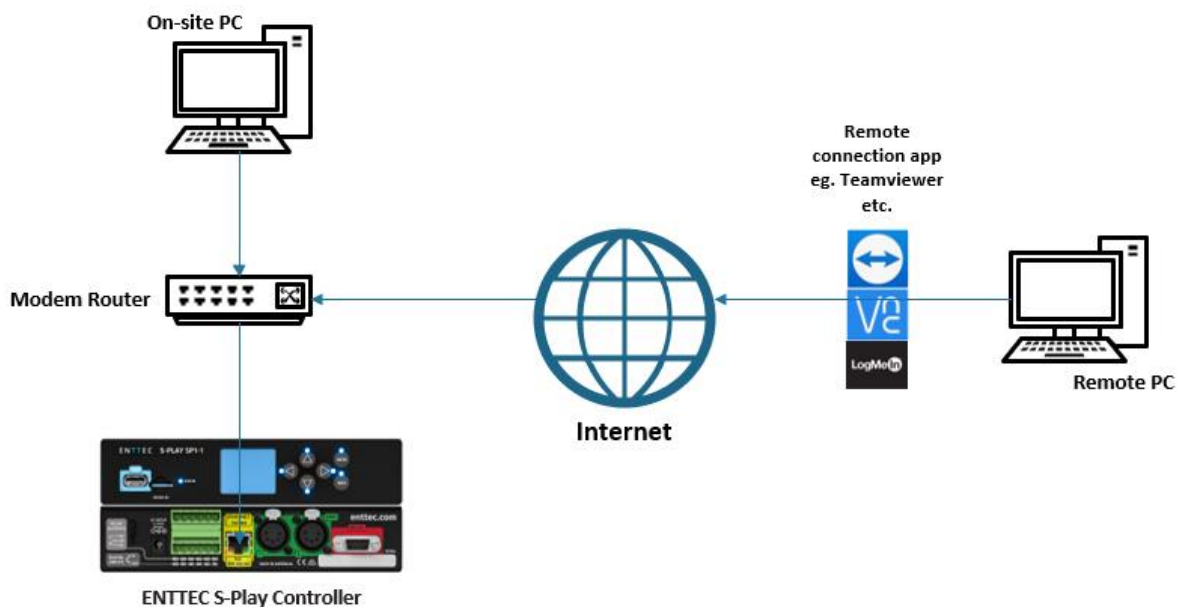


As you can see, this method uses an AWS cloud server as the intermediary to connect your device to the S-Play. This seems a bit counter-intuitive... why use a separate web server to make this connection? Why can't we just connect over the internet straight to the S-Play?

Well, we can, using the 2 methods listed before. Lets 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:



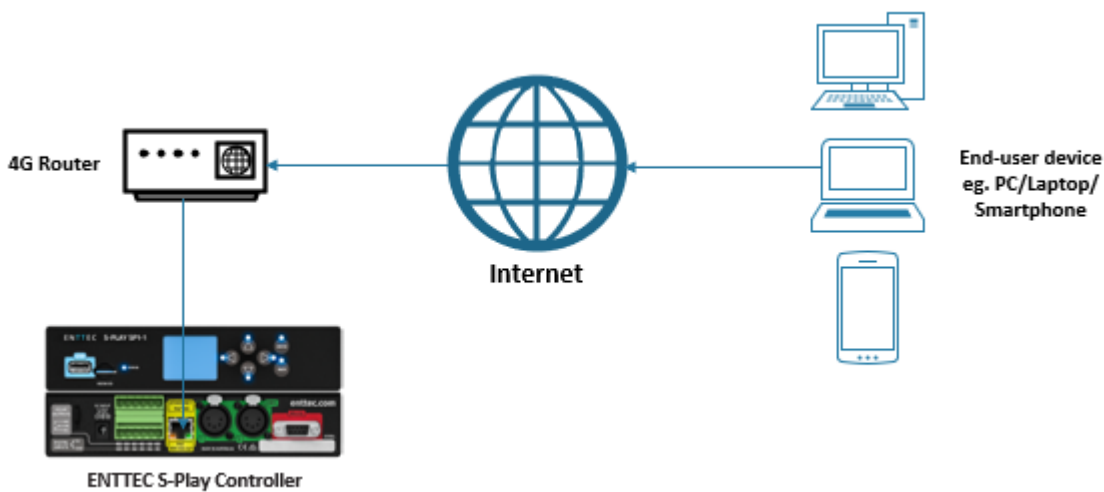
3 - Remote desktop network structure

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 a bit 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 new 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.

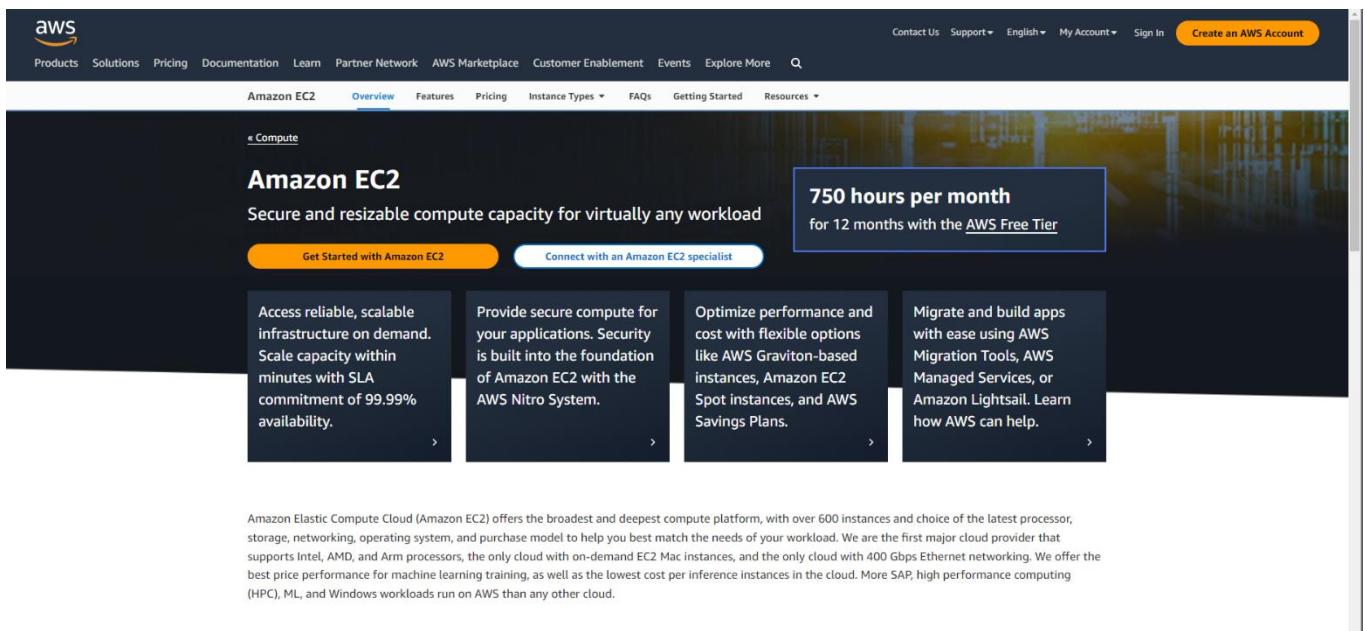
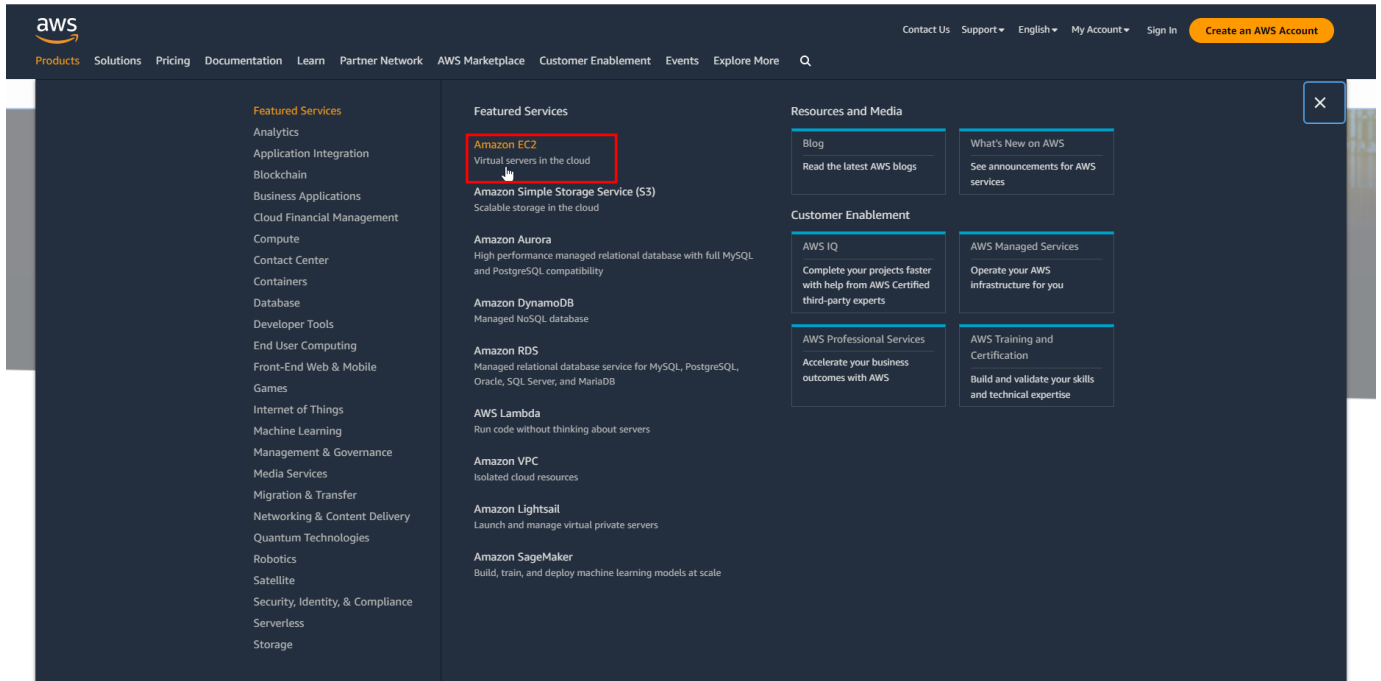
The screenshot displays the settings interface for ENTTEC S-Play. On the left is a vertical sidebar menu with the following items: INFO, OUTPUTS, ARTNET, SACN, DMX, NETWORK, SYNC, REMOTE (highlighted with a red box), DATE & TIME, LOCATION, PLAYBACK, FIRMWARE, and RESET. The main content area shows a list of settings categories: DMX settings, Network settings, Sync settings, Remote Access (expanded), Date & time settings, Location settings, Playback configuration, Firmware update, and Factory reset. The 'Remote Access' section is active and contains the following fields: 'Enable' (a toggle switch that is currently off, with a globe icon to its right), 'IP' (text input field containing '0.0.0.0'), 'Remote port' (text input field containing '8080'), 'Username' (text input field containing 'ec2-user'), and 'SSH Key' (a button labeled 'SHOW INPUT'). Below these fields is a blue 'UPDATE' button.

### 5 - S-Play remote access settings

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

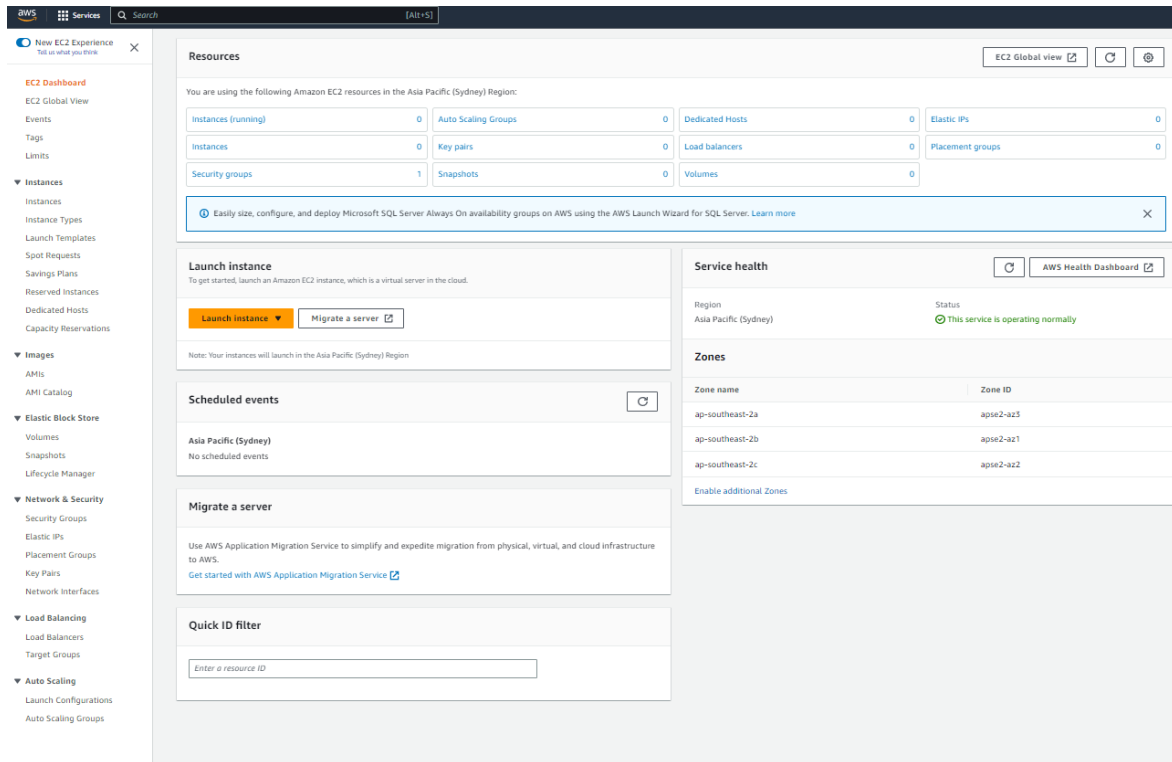
## Setting up the cloud server

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.



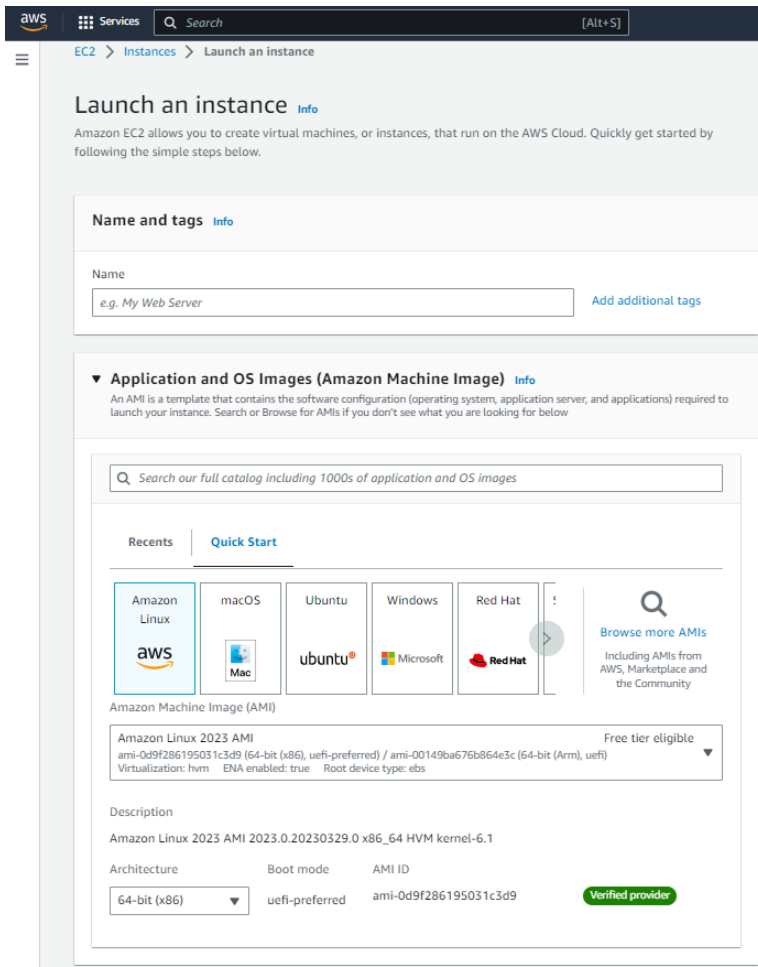
### 6 - AWS Server type selection

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



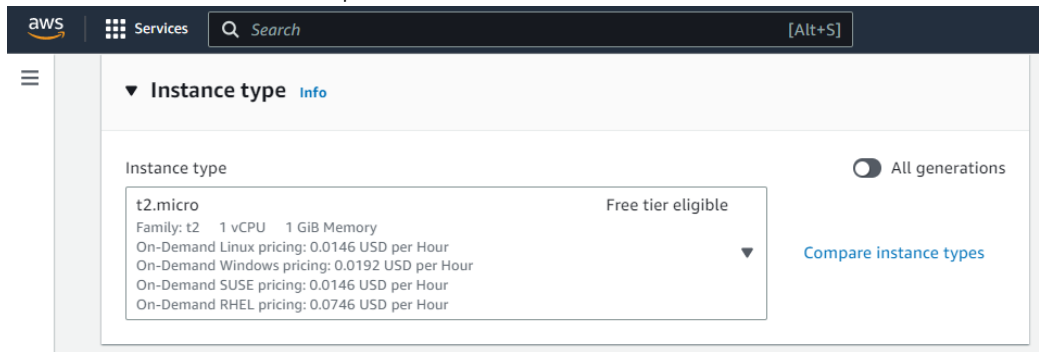
7 - AWS – Launch Instance

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

- 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.

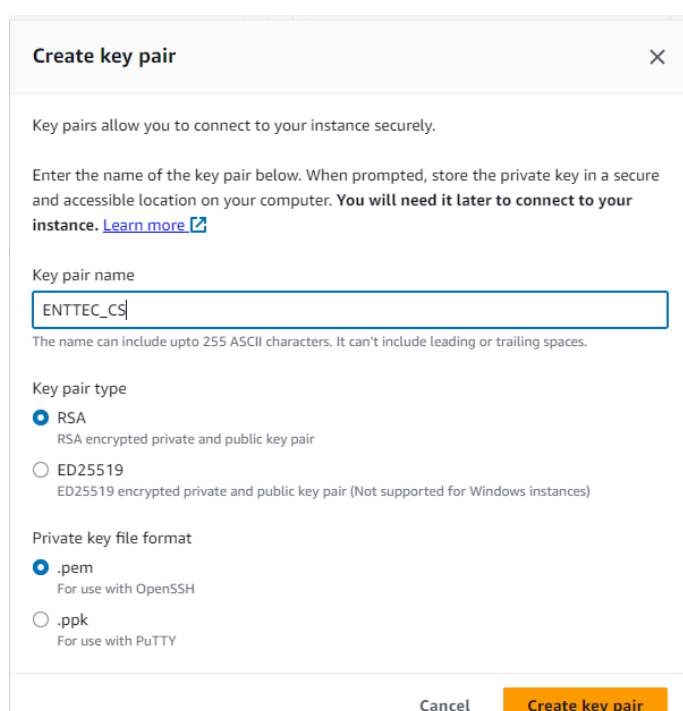
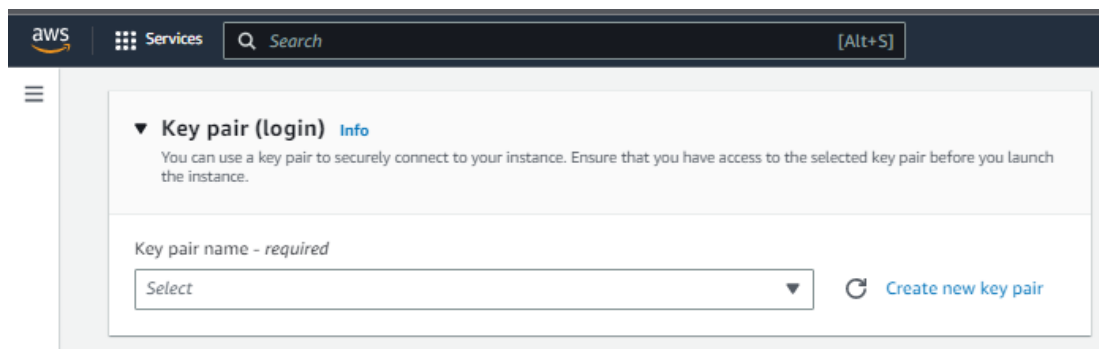


9 - AWS – Instance Type Selection

- Key pair is a key file that your S-Play will need, to be able to connect to the AWS server – remember that “SSH Key” file the S-Play was asking for?

Make sure to create a new key pair and save this in a safe location, since you won't have any way to access it again if you lose the file. The key pair will be in the form of a **.pem** file which you should upload to your S-Play in the SSH Key field.

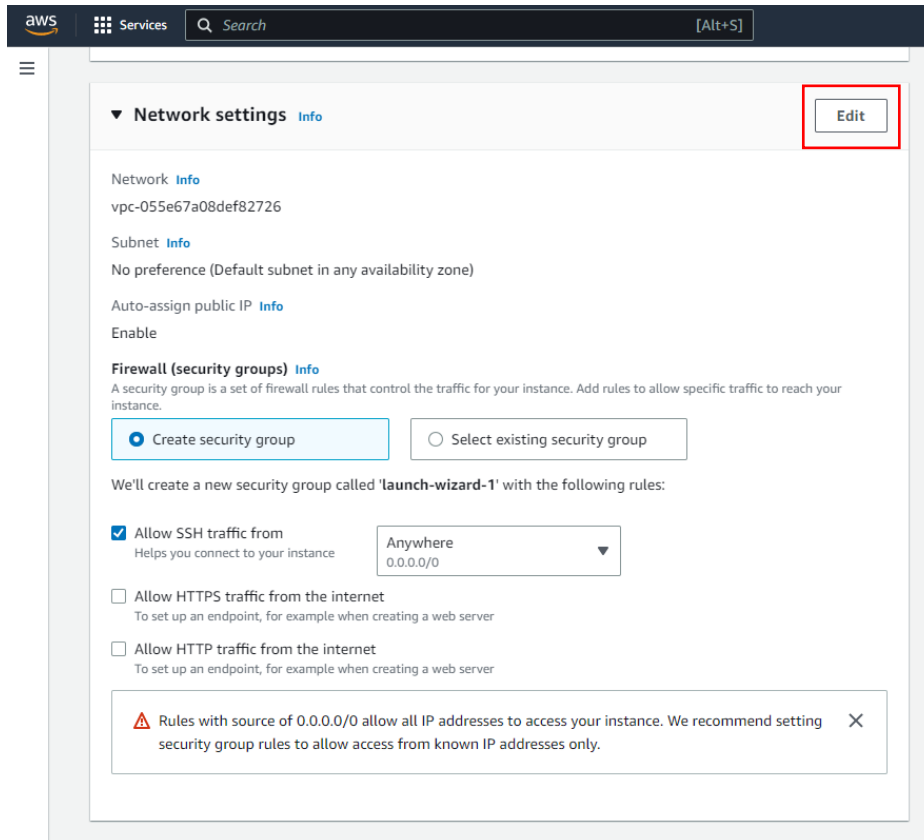
**Note:** Disabling remote access on the S-Play will wipe all SSH configuration including the **.pem** SSH Key from the S-Play for security reasons.



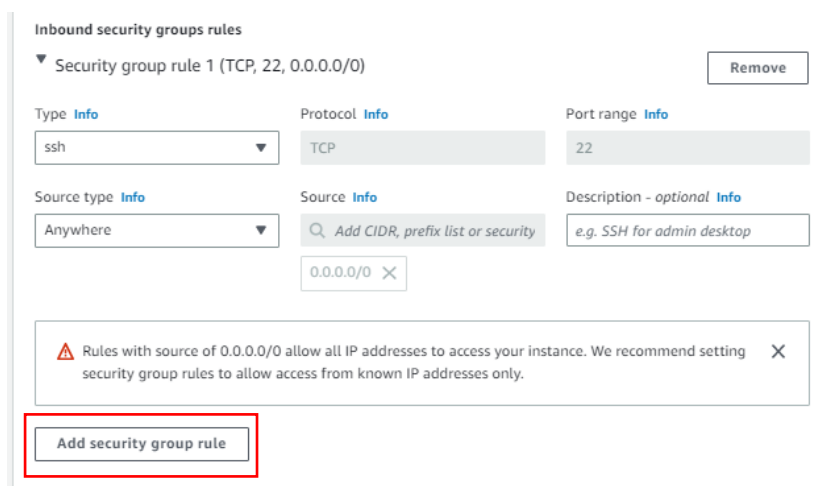
9 - AWS - key pair creation



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.



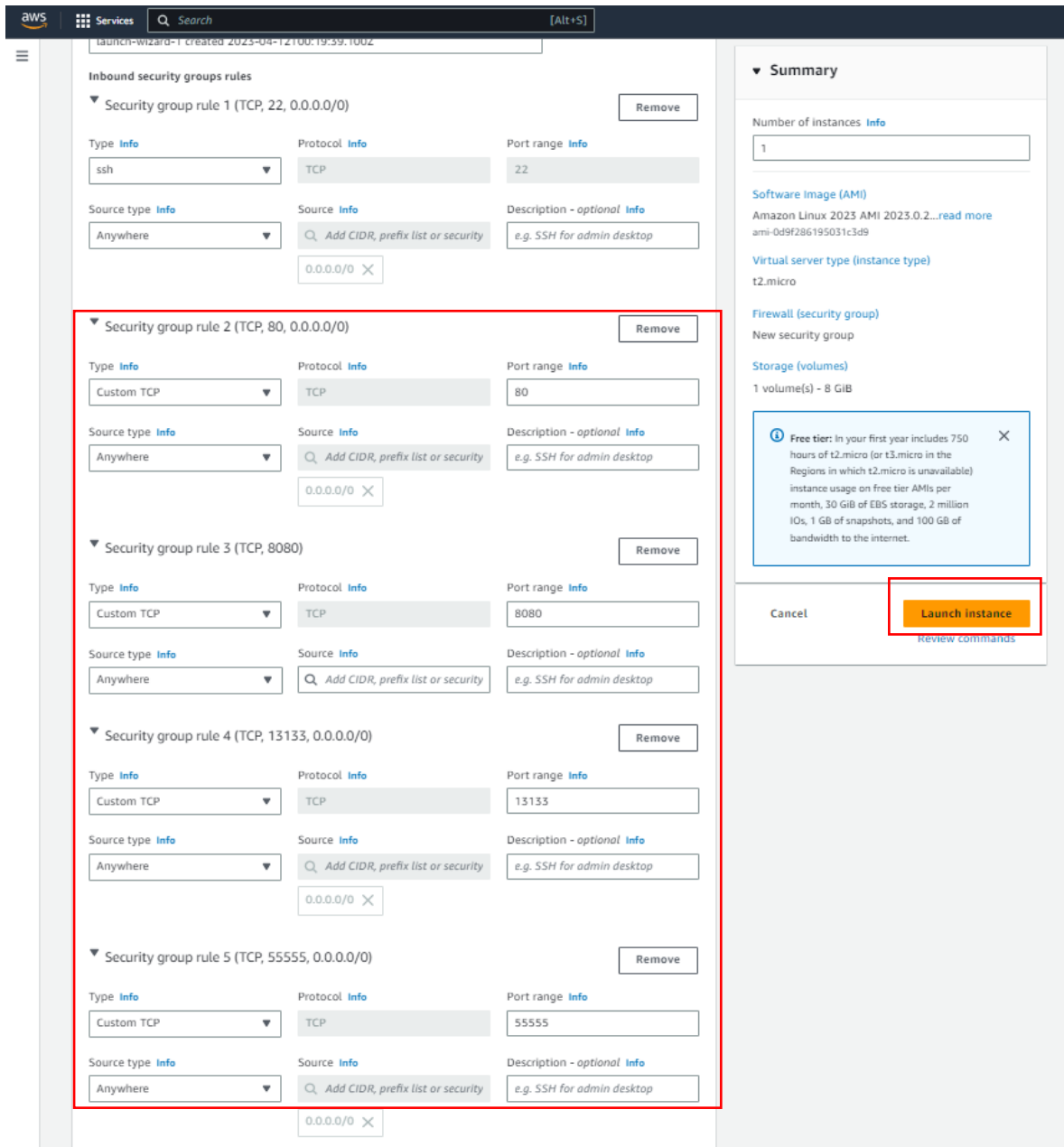
7. By default, the SSH rule (Port range 22) will be present in Inbound security groups rules, we just need to change the source to "anywhere". In addition to this we need to add exceptions for 4 additional port ranges: 80, 8080, 13133 & 55555 by clicking 'Add security group rule'. These are ports that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



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.

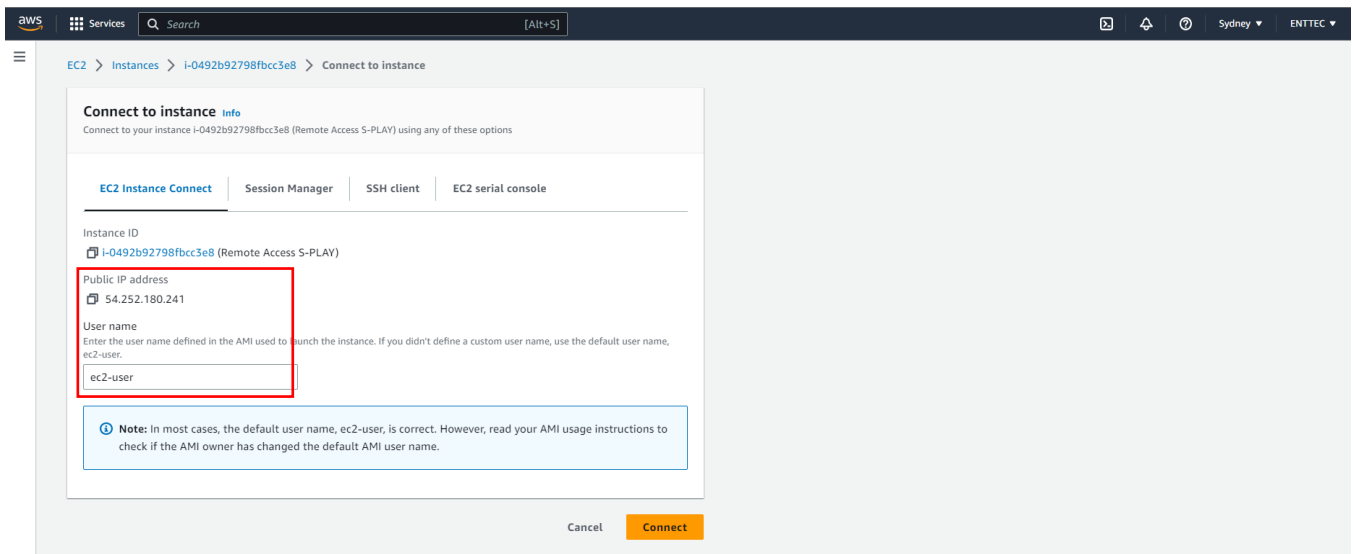
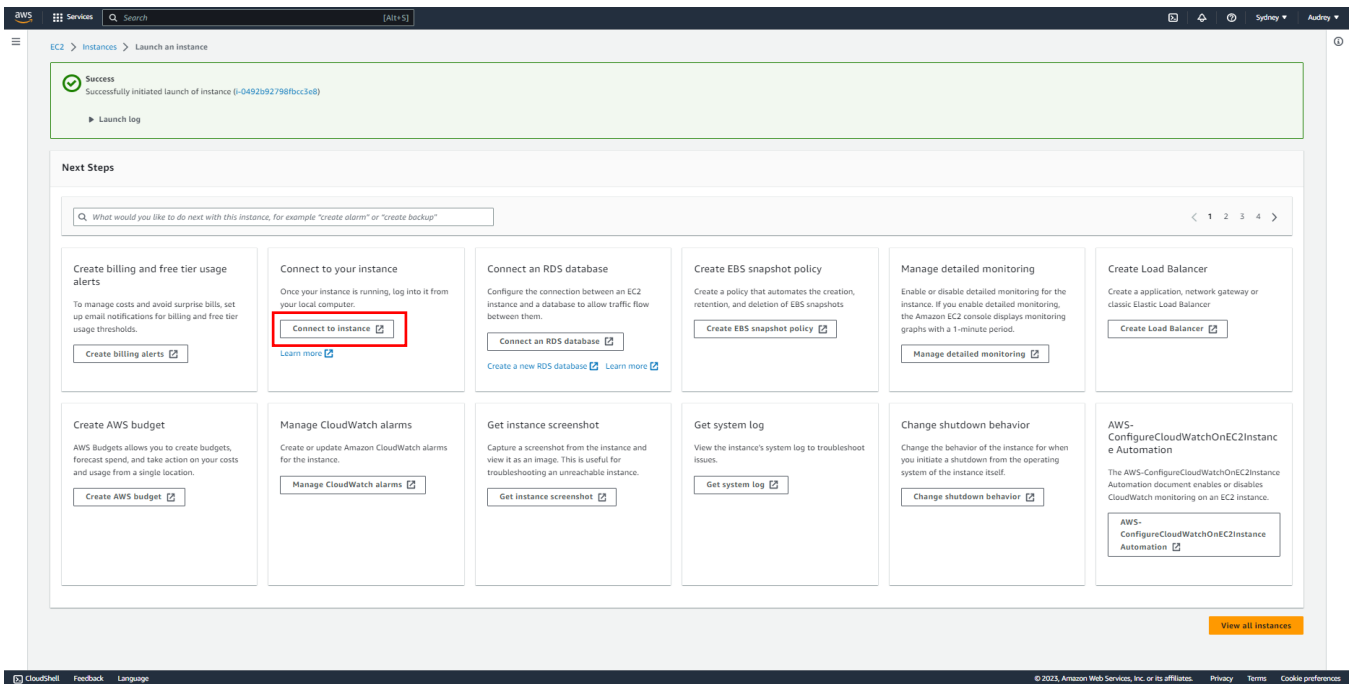
You can improve the security of the system by limiting the source ranges. For example, if you work for an integration business and you are installing S-Plays on your clients' projects. You can set the source to be the client's office IP address range. This means only a device in the client's office can 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'.



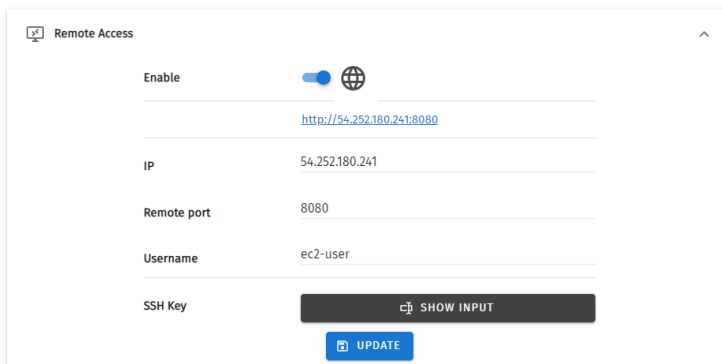
10 - AWS - port opening settings & Launch Instance

- After the instance is successfully created, we get 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.



### 11 - AWS - connect to instance

- We can enter both these values into the appropriate lines on our S-Play remote access settings and click update to ensure those are saved.



### 12 - S-Play - update remote access settings

- 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.

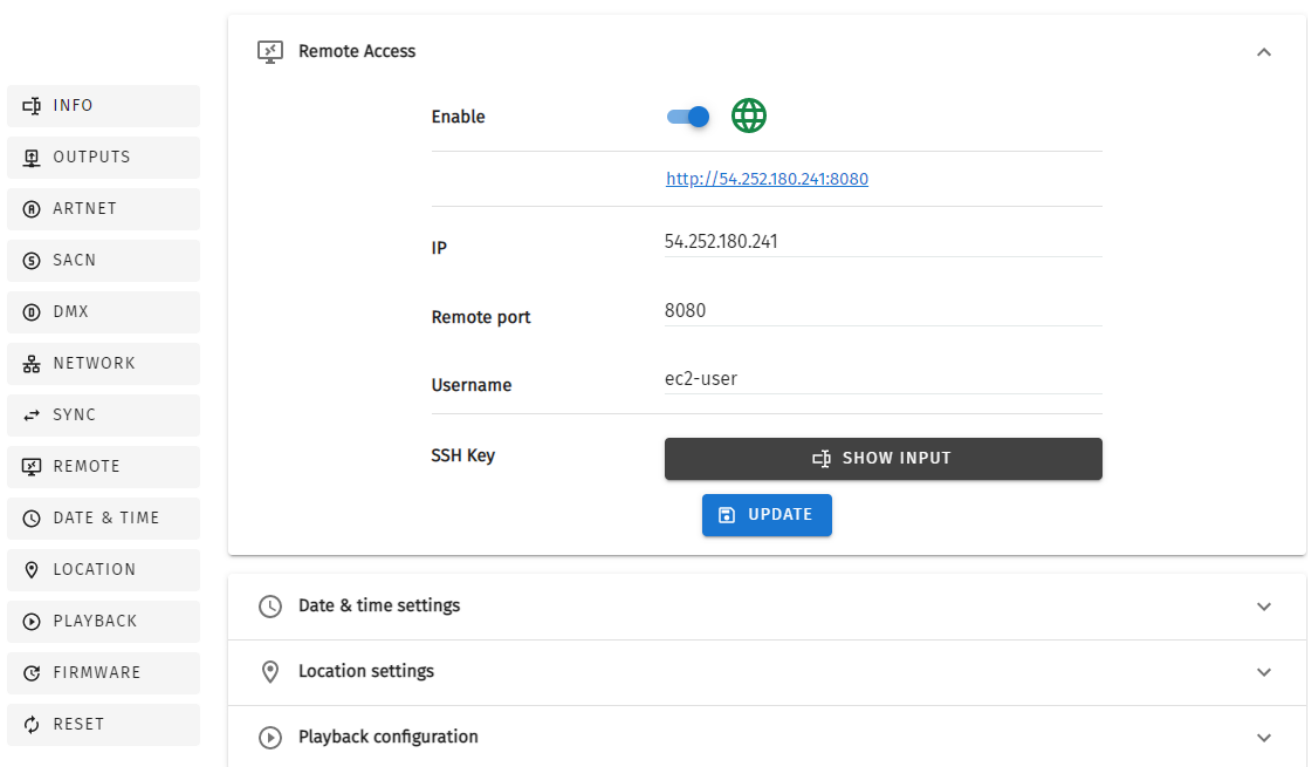
Here is the code segment you will be needing:

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



13 - AWS - server command prompt window

- After copying that code segment in, the necessary update is made, and 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. Be sure to check this URL directly from your computer, as well as from a separate device like your smart phone. If you're programming this device to go on a remote site that you don't have easy access to, then you want to make sure it's working before you leave!



14 - S-Play - successfully updated remote connection settings

## 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

That brings us to the end of this guide. 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. Just remember that this system relies on both ends – the S-Play **and** the connecting device to both have internet access. If there are disruptions at either end, this connection method won't work.

We hope you found this application note useful!

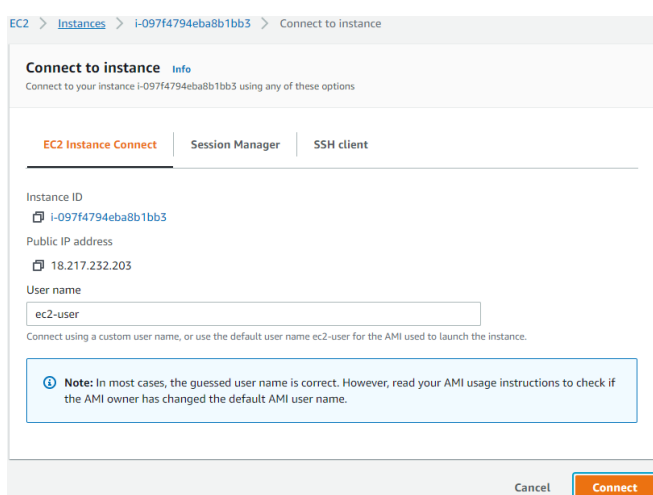
## Remote Access: Troubleshooting

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 incidence on AWS. Press 'connect 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.