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

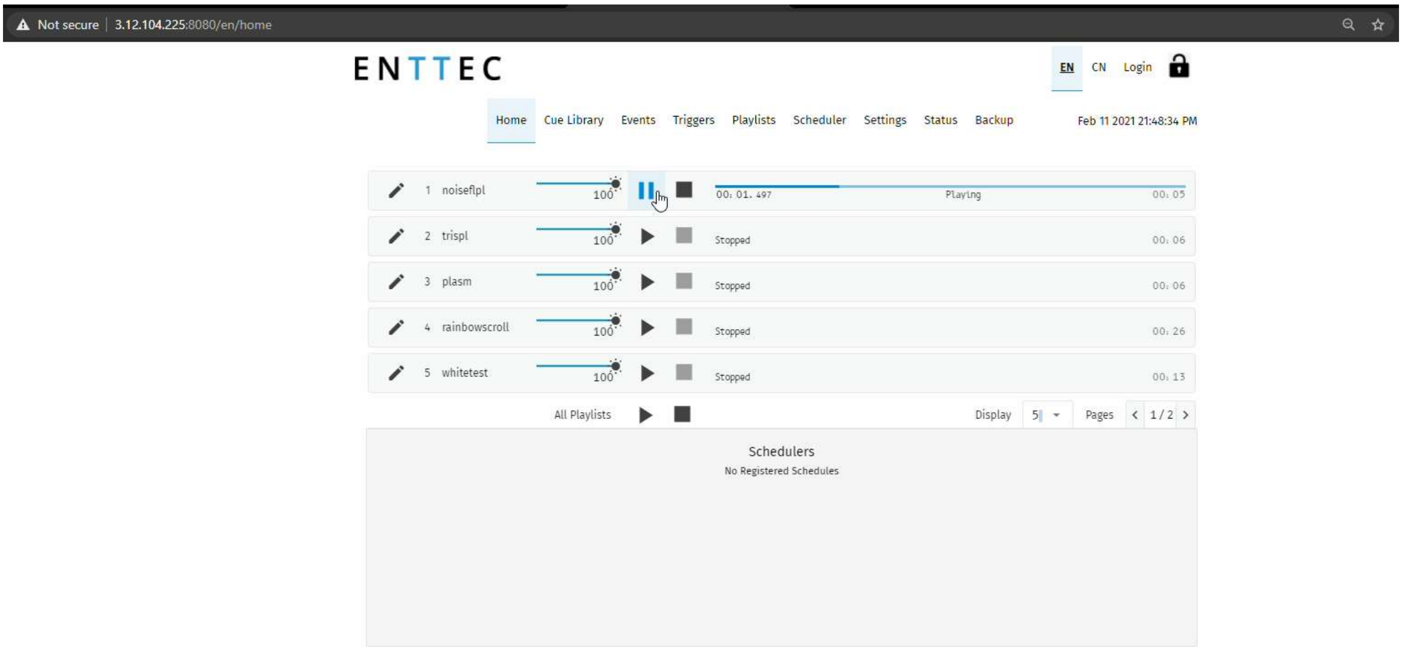
- Before connecting your S-Play to the Internet ensure your local network firewall provides security all devices have been adequately secured.
- If ever unsure consult a qualified professional.
- Ensure you have sufficient extra bandwidth to deal with influxes of traffic caused by an internet connection.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



3.12.104.225:8080/en/home

Feb 11 2021 21:48:34 PM

1	noiseflpl	100	00: 01. 497 Playing	00: 05
2	trispl	100	Stopped	00: 06
3	plasm	100	Stopped	00: 06
4	rainbowscroll	100	Stopped	00: 26
5	whitetest	100	Stopped	00: 13

Schedulers
No Registered Schedules

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.

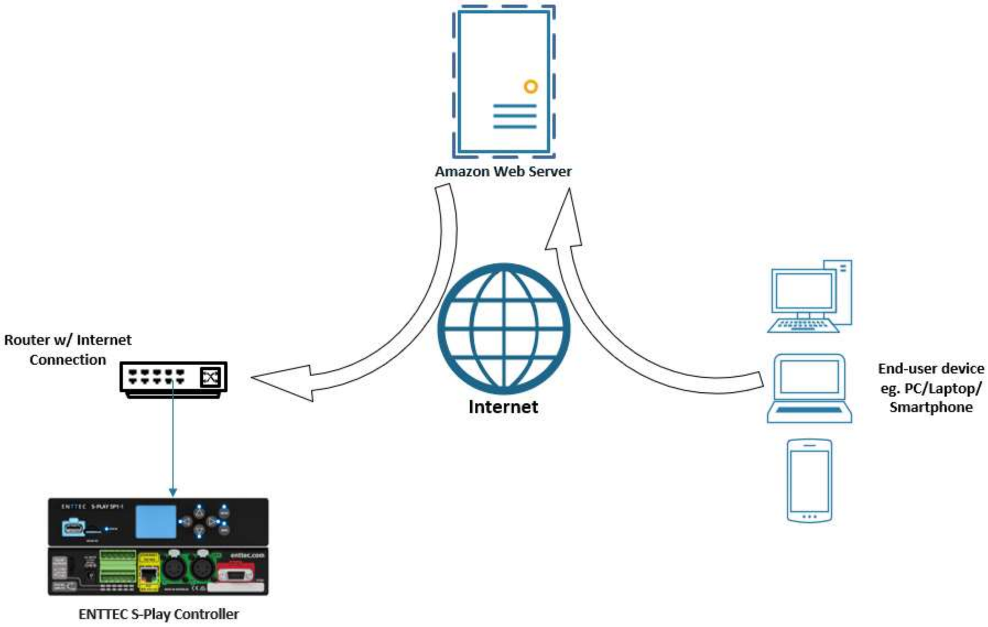
Remote connection options + network structure

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:

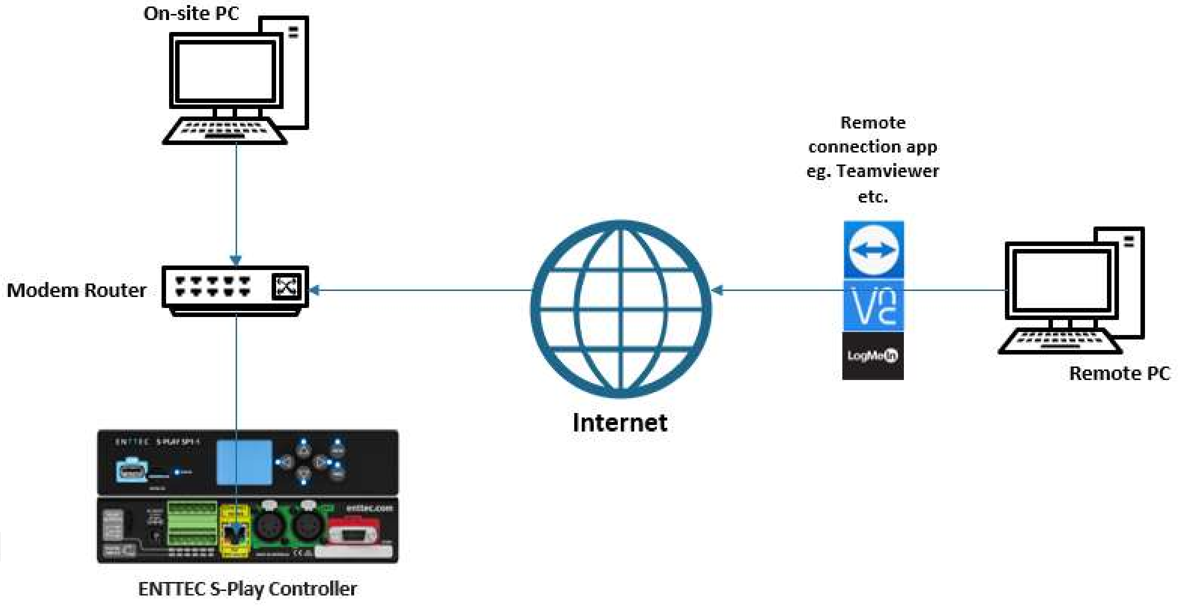

Amazon Web Server
Router w/ Internet Connection
Internet
End-user device eg. PC/Laptop/ Smartphone
ENTTEC S-Play Controller

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:



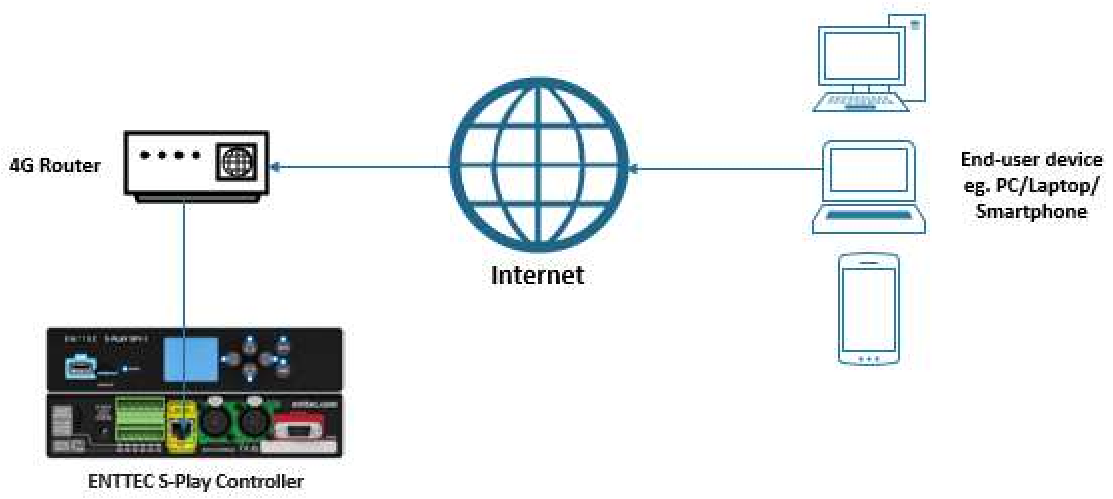
3 - Remote desktop network structure

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



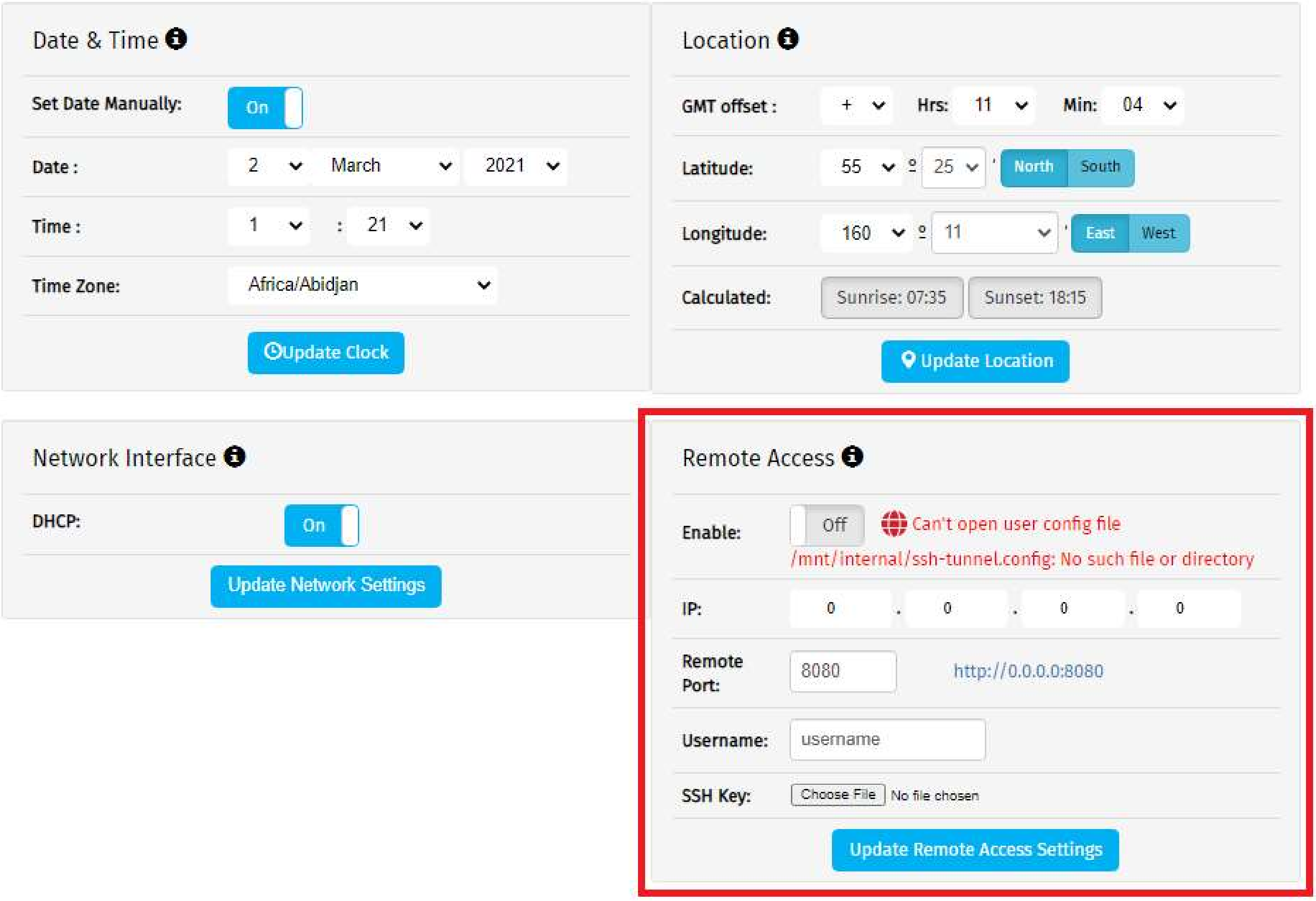
4 - Port forwarding network structure

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, once we scroll down to near the bottom, we'll see a new section titled: "Remote SSH 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.



Date & Time

Set Date Manually: On

Date : 2 March 2021

Time : 1 : 21

Time Zone: Africa/Abidjan

Location

GMT offset : + Hrs: 11 Min: 04

Latitude: 55 º 25 ' North South

Longitude: 160 º 11 ' East West

Calculated: Sunrise: 07:35 Sunset: 18:15

Network Interface

DHCP: On

Remote Access

Enable: Off Can't open user config file /mnt/internal/ssh-tunnel.config: No such file or directory

IP: 0 . 0 . 0 . 0

Remote Port: 8080 http://0.0.0.0:8080

Username: username

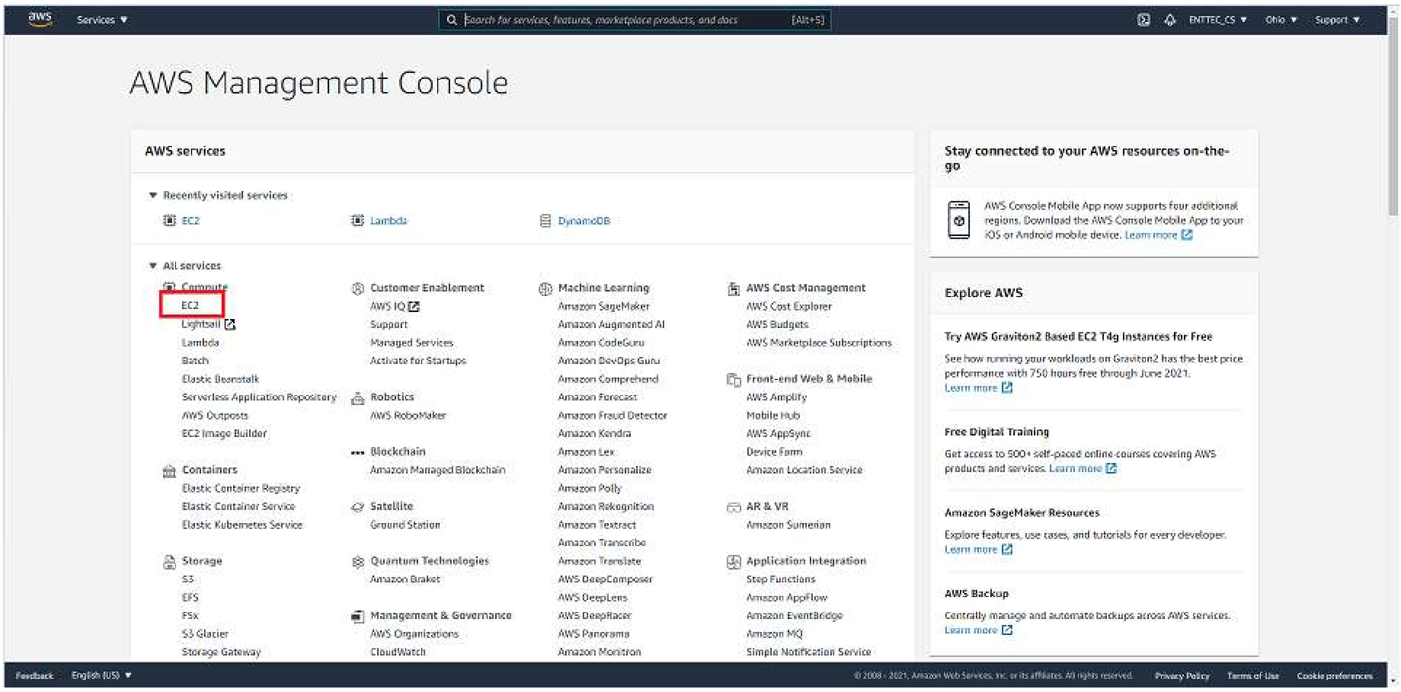
SSH Key: Choose File No file chosen

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.


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
ENTTEC_CS ▼
Ohio ▼
Support ▼

AWS Management Console

AWS services

▼ Recently visited services

EC2 Lambda DynamoDB

▼ All services

Compute
EC2
Lightsail
Lambda
Batch
Elastic Beanstalk
Serverless Application Repository
AWS Outposts
EC2 Image Builder

Containers
Elastic Container Registry
Elastic Container Service
Elastic Kubernetes Service

Storage
S3
EFS
FSx
S3 Glacier
Storage Gateway

Customer Enablement
AWS IQ
Support
Managed Services
Activate for Startups

Robotics
AWS RoboMaker

Blockchain
Amazon Managed Blockchain

Satellite
Ground Station

Quantum Technologies
Amazon Braket

Management & Governance
AWS Organizations
CloudWatch

Machine Learning
Amazon SageMaker
Amazon Augmented AI
Amazon CodeGuru
Amazon DevOps Guru
Amazon Comprehend
Amazon Forecast
Amazon Fraud Detector
Amazon Kendra
Amazon Lex
Amazon Personalize
Amazon Polly
Amazon Rekognition
Amazon Textract
Amazon Transcribe
Amazon Translate
AWS DeepComposer
AWS DeepLens
AWS DeepRacer
AWS Panorama
Amazon Monitron

AWS Cost Management
AWS Cost Explorer
AWS Budgets
AWS Marketplace Subscriptions

Front-end Web & Mobile
AWS Amplify
Mobile Hub
AWS AppSync
Device Farm
Amazon Location Service

AR & VR
Amazon Sumerian

Application Integration
Step Functions
Amazon AppFlow
Amazon EventBridge
Amazon MQ
Simple Notification Service

Stay connected to your AWS resources on-the-go

AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

Explore AWS

Try AWS Graviton2 Based EC2 T4g Instances for Free

See how running your workloads on Graviton2 has the best price performance with 750 hours free through June 2021. Learn more

Free Digital Training

Get access to 500+ self-paced online courses covering AWS products and services. Learn more

Amazon SageMaker Resources

Explore features, use cases, and tutorials for every developer. Learn more

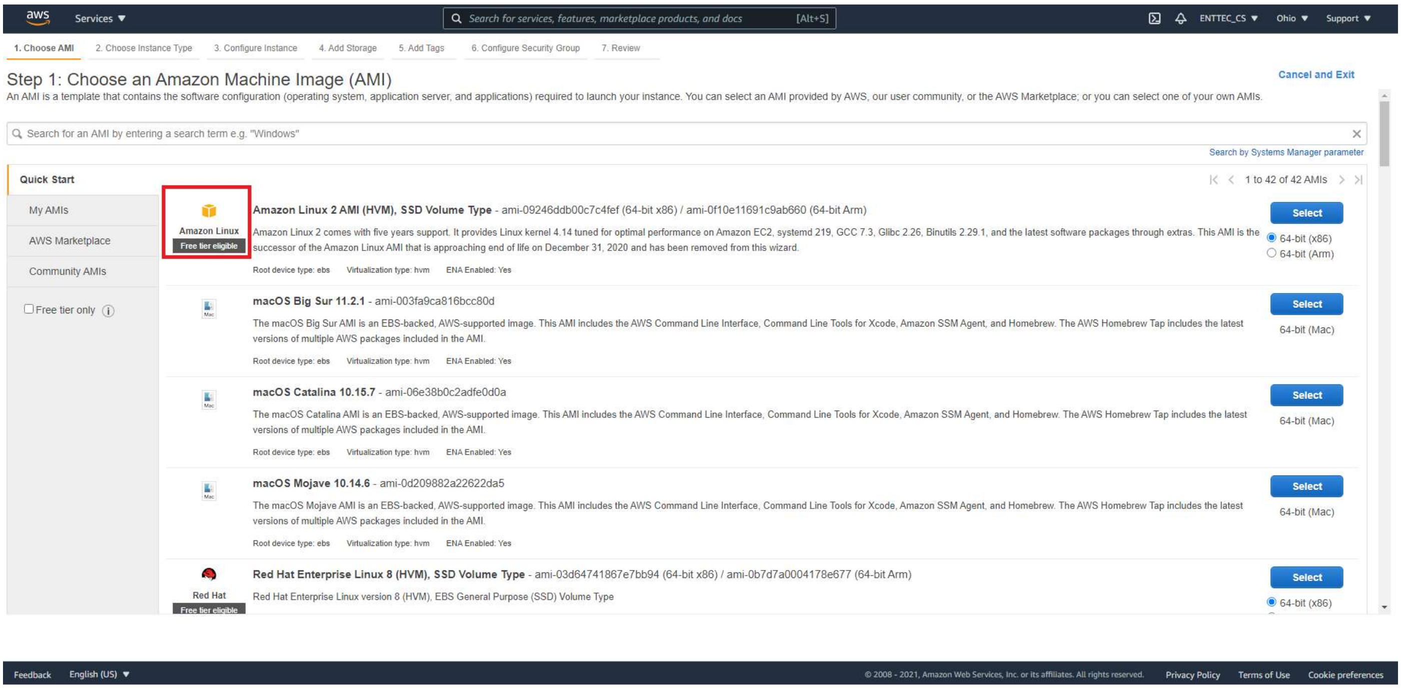
AWS Backup

Centrally manage and automate backups across AWS services. Learn more

Feedback
English (US) ▼
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

6 - AWS server type selection

2. Next, we'll go to "Launch Instance" to create a new server instance. For the Amazon Machine Image, we'll just chose Amazon Linux, again because it's free tier.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Amazon Linux — Free tier eligible

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-09246ddb00c7c4fef (64-bit x86) / ami-0f10e11691c9ab660 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select ● 64-bit (x86) ○ 64-bit (Arm)

macOS Big Sur 11.2.1 - ami-003fa9ca816bcc80d

The macOS Big Sur AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select 64-bit (Mac)

macOS Catalina 10.15.7 - ami-06e38b0c2adfe0d0a

The macOS Catalina AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select 64-bit (Mac)

macOS Mojave 10.14.6 - ami-0d209882a22622da5

The macOS Mojave AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select 64-bit (Mac)

Red Hat — Free tier eligible

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-03d64741867e7bb94 (64-bit x86) / ami-0b7d7a0004178e677 (64-bit Arm)

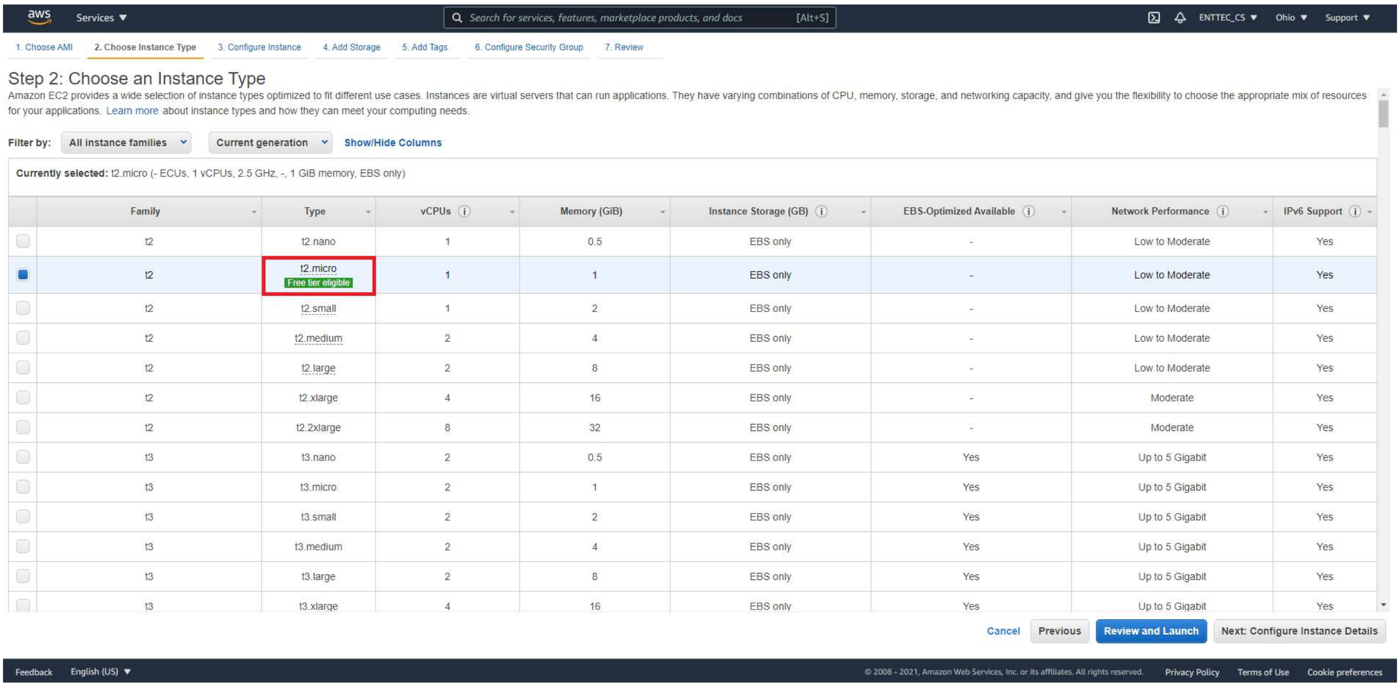
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Select ● 64-bit (x86)

Feedback
English (US) ▼
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

7 - AWS - Machine image selection

3. 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.



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance families | Current generation | Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

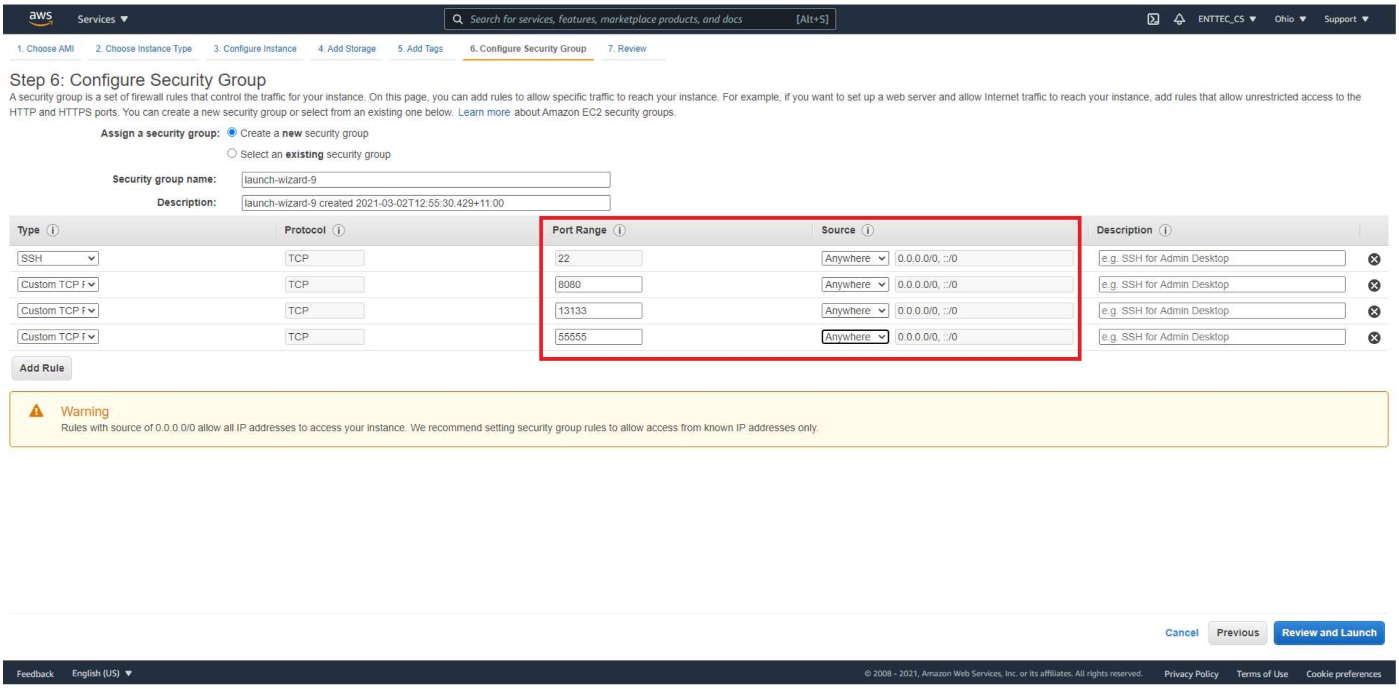
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
□	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
□	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
□	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
□	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
□	t3	t3.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel
Previous
Review and Launch
Next: Configure Instance Details
Feedback
English (US) ▼
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

8 - AWS - Instance selection

4. No need to change anything in "Instance Details", no need to "Add Storage" or "Add Tags" but we can progress through to "Configure Security Group" which is where we add our port settings so that any device can access the server.

5. By default, the SSH rule (port22) will be present, we just need to change the source to "anywhere". In addition to this we need to add exceptions for 3 additional ports: 8080, 13133 & 55555. These are ports that the S-Play will be using to display its web page and allow interactivity to the connecting device.



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: launch-wizard-9

Description: launch-wizard-9 created 2021-03-02T12:55:30.429+11:00

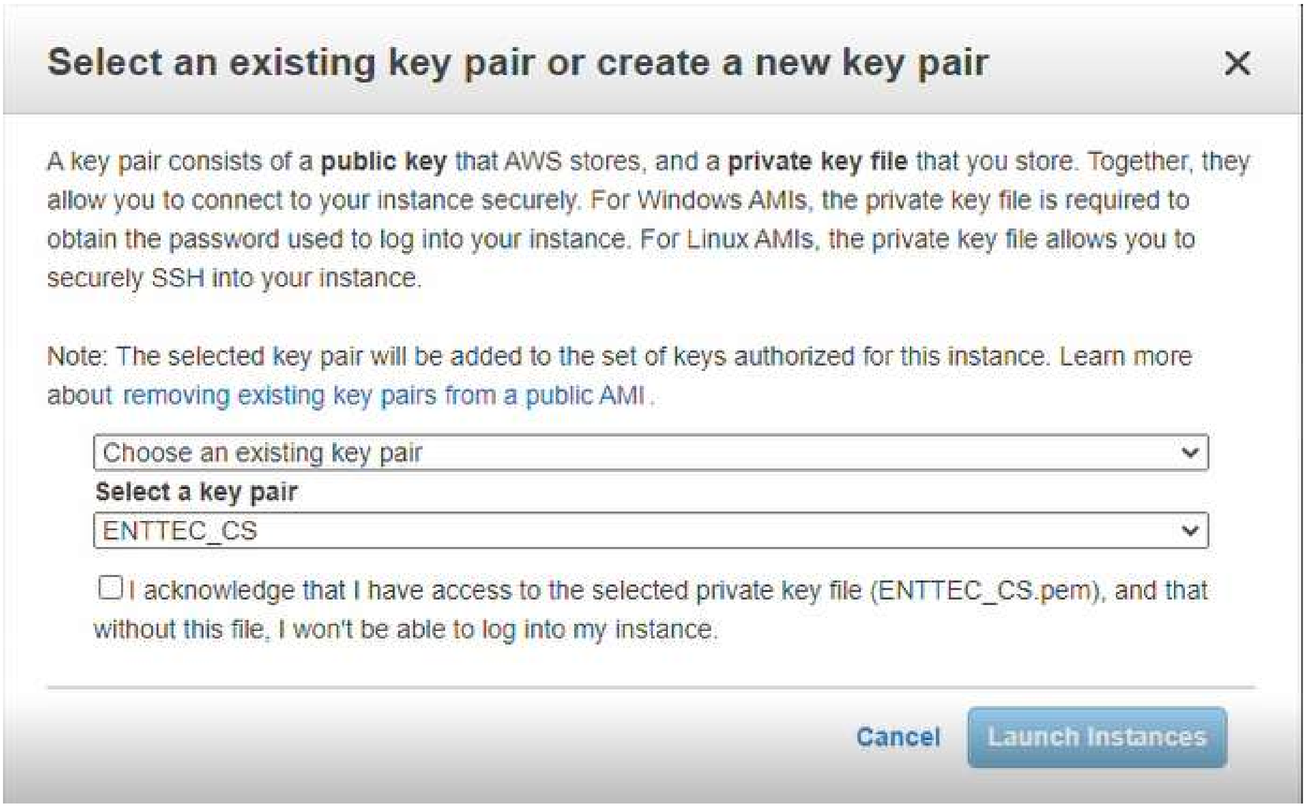
Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8080	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	13133	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	55555	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

9 - AWS - port opening settings

6. Note we have set these rules to “anywhere” 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.

7. With these rules set, we can now launch our server instance. AWS will bring up a prompt about key pairs. This 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?



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

Choose an existing key pair

Select a key pair

ENTTEC_CS

□ I acknowledge that I have access to the selected private key file (ENTTEC_CS.pem), and that without this file, I won't be able to log into my instance.

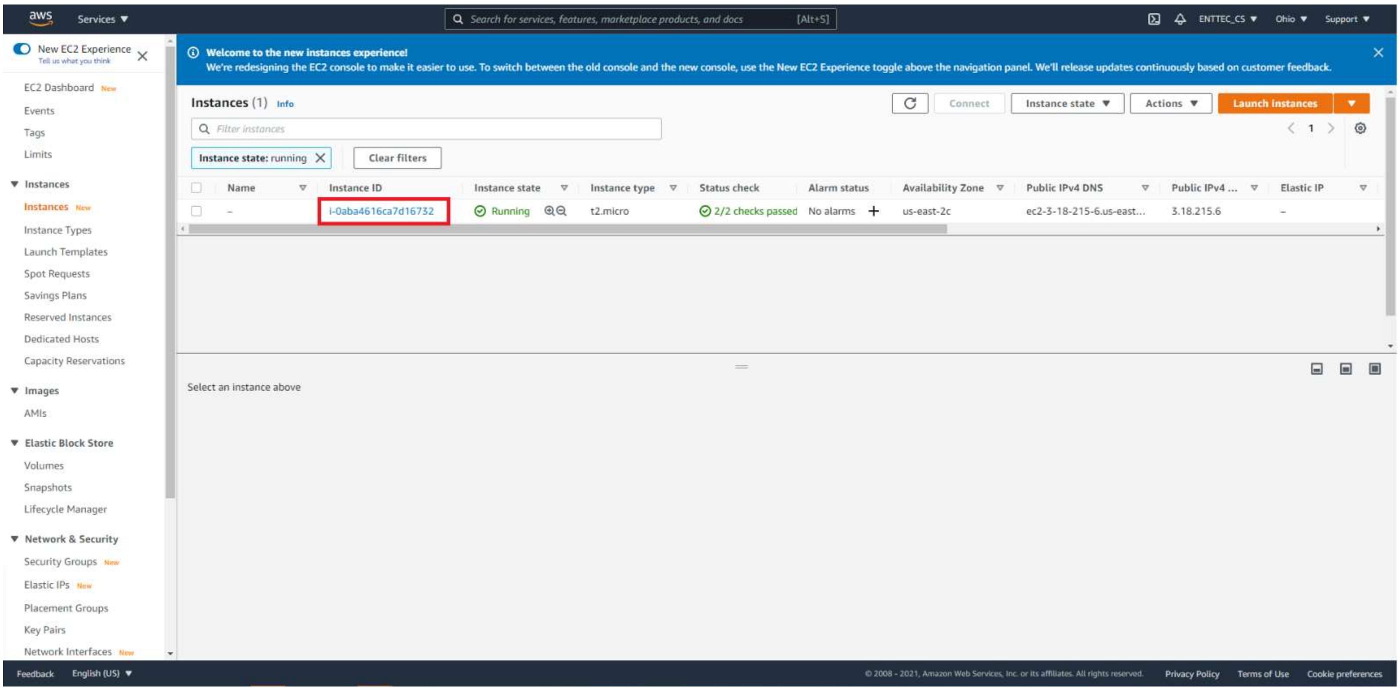
10 - AWS - key pair creation

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.

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

8. Now we can launch our server instance and view its status. This will bring up a list of instances so be sure to go to whichever one you just created. If this is your first-time using AWS then it should just be one instance that appears.



Welcome to the new instances experience!
We're redesigning the EC2 console to make it easier to use. To switch between the old console and the new console, use the New EC2 Experience toggle above the navigation panel. We'll release updates continuously based on customer feedback.

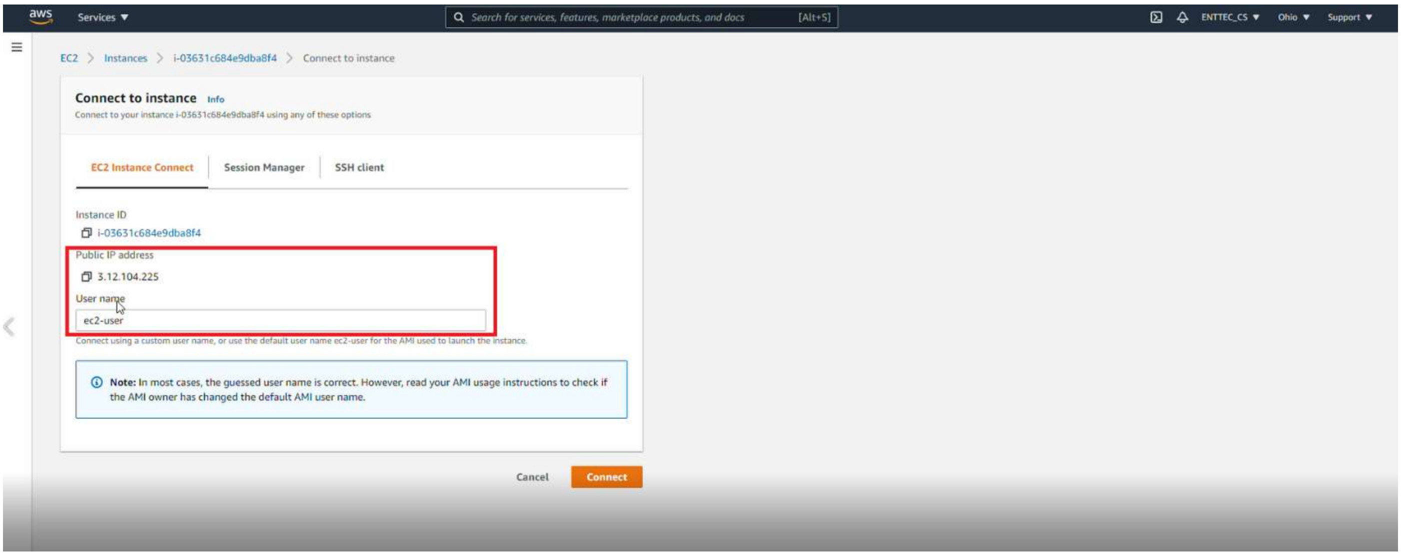
Instances (1) Info

Instance state: running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
□	–	i-0aba4616ca7d16732	Running	t2.micro	2/2 checks passed	No alarms	us-east-2c	ec2-3-18-215-6.us-east...	3.18.215.6	–

11 - AWS - Launch Instance

9. After clicking on our newly created instance, we get to the "Instance Summary" screen. From here we click "Connect" which brings us to this screen showing us the public IP address and Username of our instance. We can enter both these values into the appropriate lines on our S-Play remote connection settings and click update to ensure those are saved.


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
ENTTEC_CS ▼
Ohio ▼
Support ▼
EC2 > Instances > i-03631c684e9dba8f4 > Connect to instance
EC2 Instance Connect
Session Manager
SSH client
Cancel
Connect

Connect to instance Info
Connect to your instance i-03631c684e9dba8f4 using any of these options

Instance ID
i-03631c684e9dba8f4

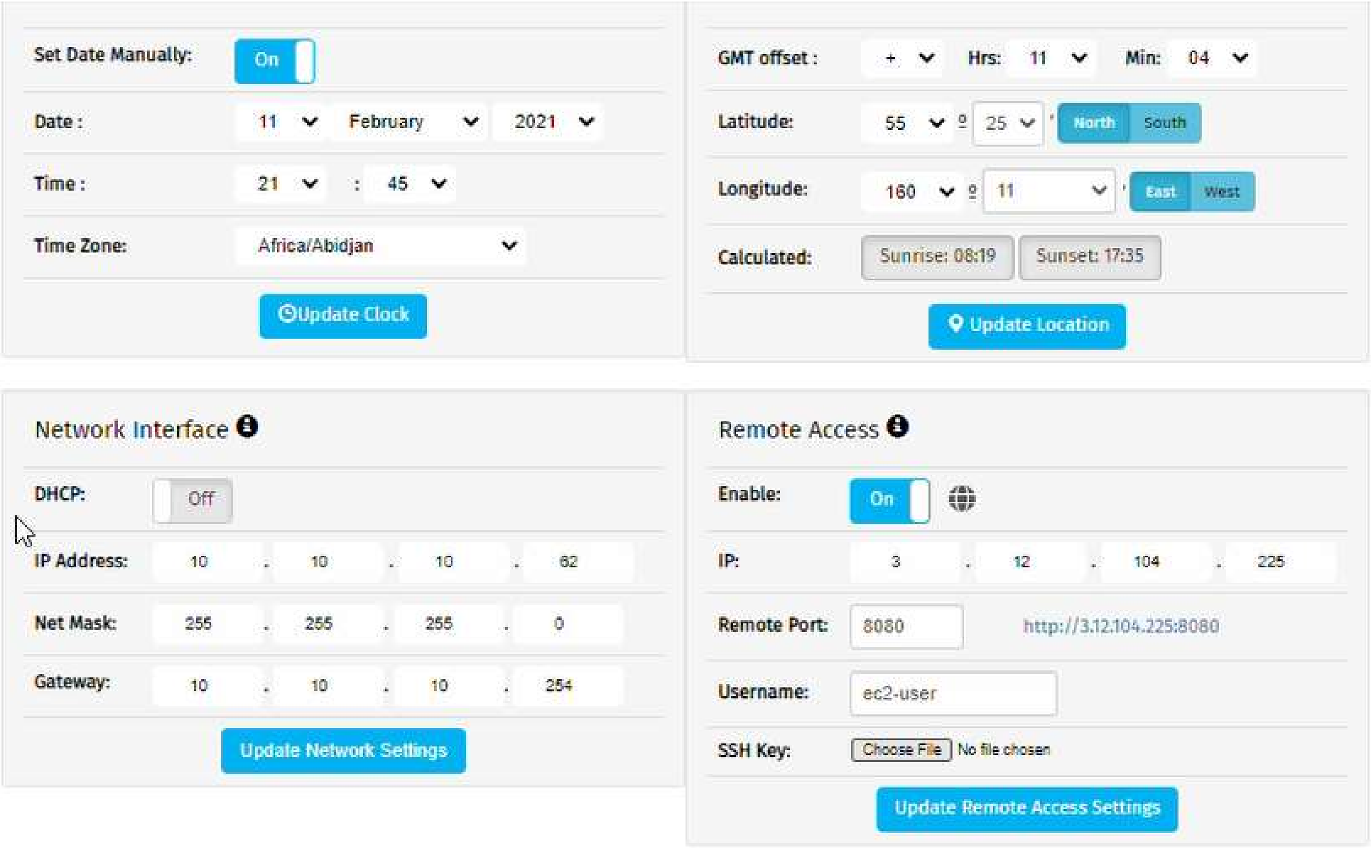
Public IP address
3.12.104.225

User name
ec2-user

Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

12 - AWS - connect to instance



Set Date Manually: On

Date : 11 February 2021

Time : 21 : 45

Time Zone: Africa/Abidjan

GMT offset : + Hrs: 11 Min: 04

Latitude: 55 º 25 ' North South

Longitude: 160 º 11 ' East West

Calculated: Sunrise: 08:19 Sunset: 17:35

Network Interface

DHCP: Off

IP Address: 10 . 10 . 10 . 62

Net Mask: 255 . 255 . 255 . 0

Gateway: 10 . 10 . 10 . 254

Remote Access

Enable: On

IP: 3 . 12 . 104 . 225

Remote Port: 8080 http://3.12.104.225:8080

Username: ec2-user

SSH Key: Choose File No file chosen

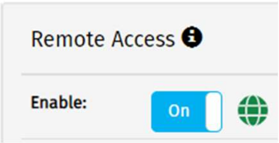
13 - S-Play - update remote access settings

10. 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:

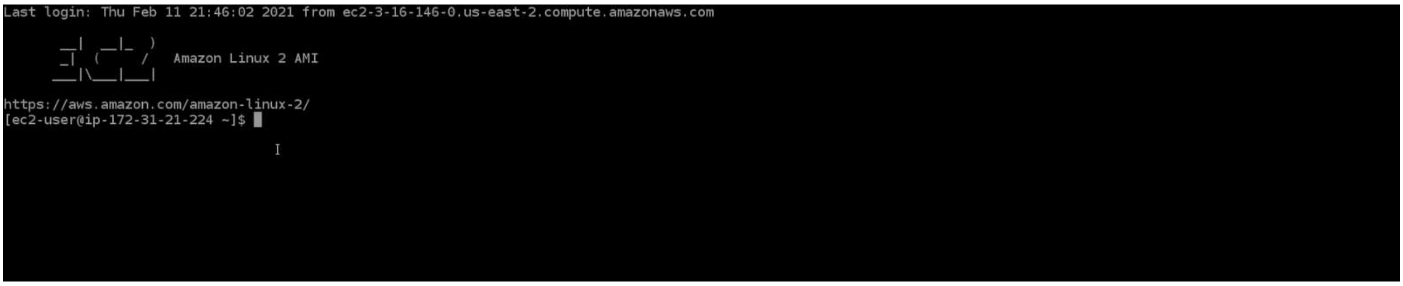
    ```
    /usr/bin/sudo /usr/bin/sed -i -e 's/.*GatewayPorts.*/GatewayPorts yes/g' /etc/ssh/sshd_config
    /usr/bin/sudo /usr/bin/systemctl restart sshd
    ```

11. 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!



Remote Access

Enable: On

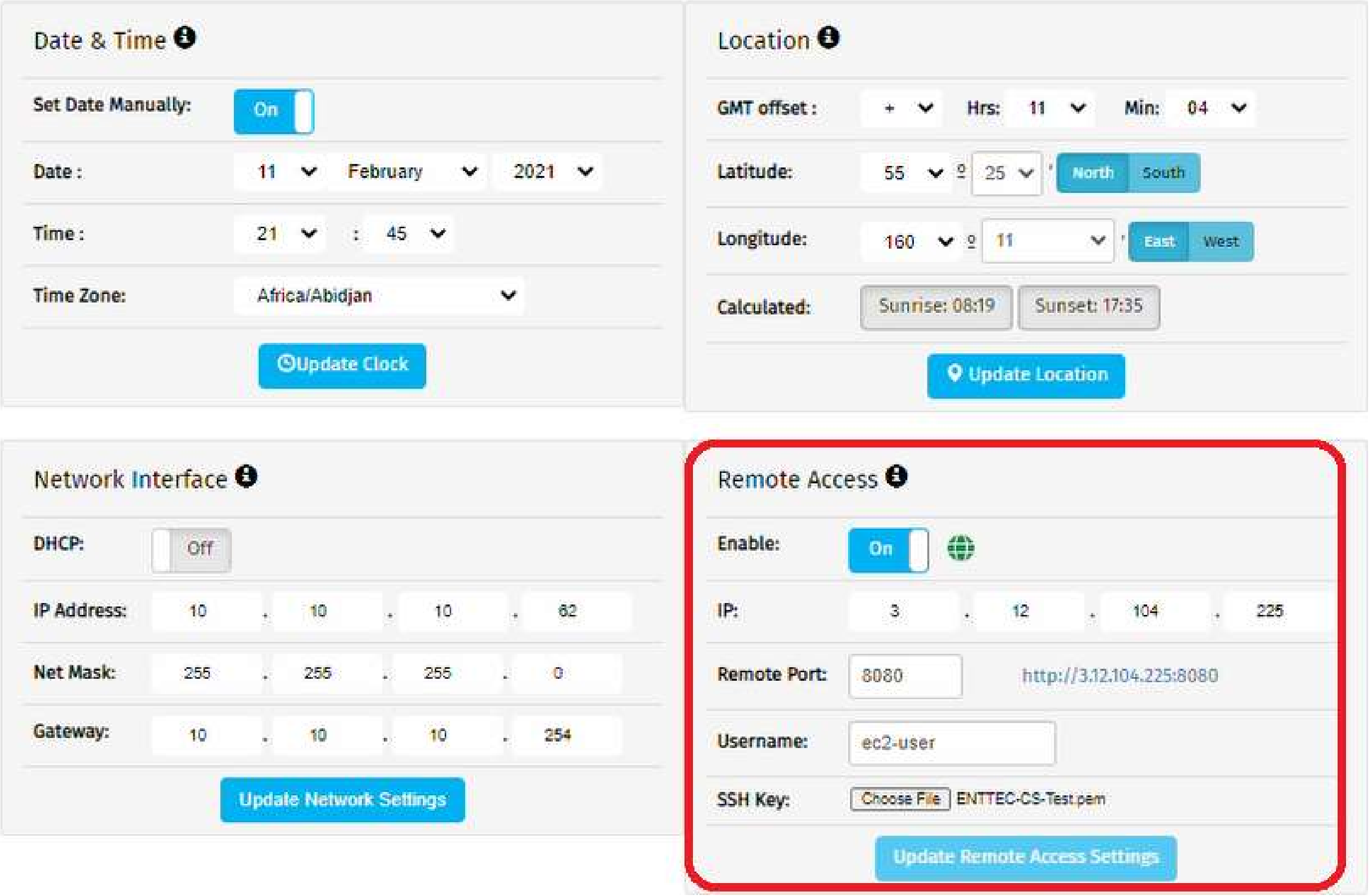


```
Last login: Thu Feb 11 21:46:02 2021 from ec2-3-16-146-0.us-east-2.compute.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-21-224 ~]$ █
```

14 - AWS - server command prompt window



Date & Time
Set Date Manually: On
Date : 11 February 2021
Time : 21 : 45
Time Zone: Africa/Abidjan

Location
GMT offset : + Hrs: 11 Min: 04
Latitude: 55 º 25 ' North South
Longitude: 160 º 11 ' East West
Calculated: Sunrise: 08:19 Sunset: 17:35

Network Interface
DHCP: Off
IP Address: 10 . 10 . 10 . 62
Net Mask: 255 . 255 . 255 . 0
Gateway: 10 . 10 . 10 . 254

Remote Access
Enable: On
IP: 3 . 12 . 104 . 225
Remote Port: 8080 http://3.12.104.225:8080
Username: ec2-user
SSH Key: Choose File ENTTEC-CS-Test.pem

15 - 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

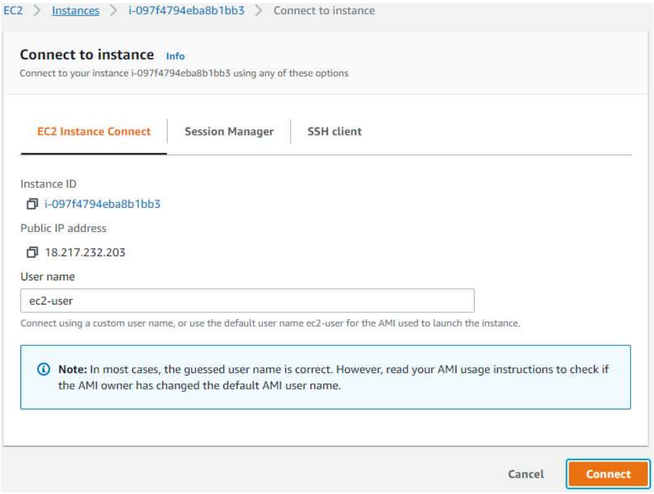
On first connection, the S-Play will automatically try to configure the server's gateway by running:

```
/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.



Connect to instance Info

Connect to your instance i-097f4794eba8b1bb3 using any of these options

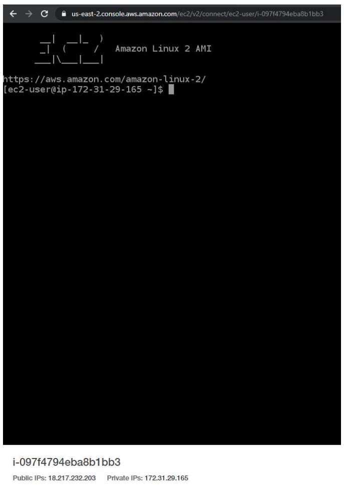
Instance ID
i-097f4794eba8b1bb3

Public IP address
18.217.232.203

User name
ec2-user

Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.


us-east-2.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-097f4794eba8b1bb3

```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-29-165 ~]$ 
```

i-097f4794eba8b1bb3

Public IPs: 18.217.232.203   Private IPs: 172.31.29.165

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.