

# ExpressCluster AWS Witness Server

How-to setup an ExpressCluster witness server using an AWS Lightsail instance.

## Prerequisites

- [Amazon AWS](#) account
- An existing cluster running NEC ExpressCluster 4.1 or higher
- [WinSCP](#)
- [Putty](#) or other SSH client (optional)

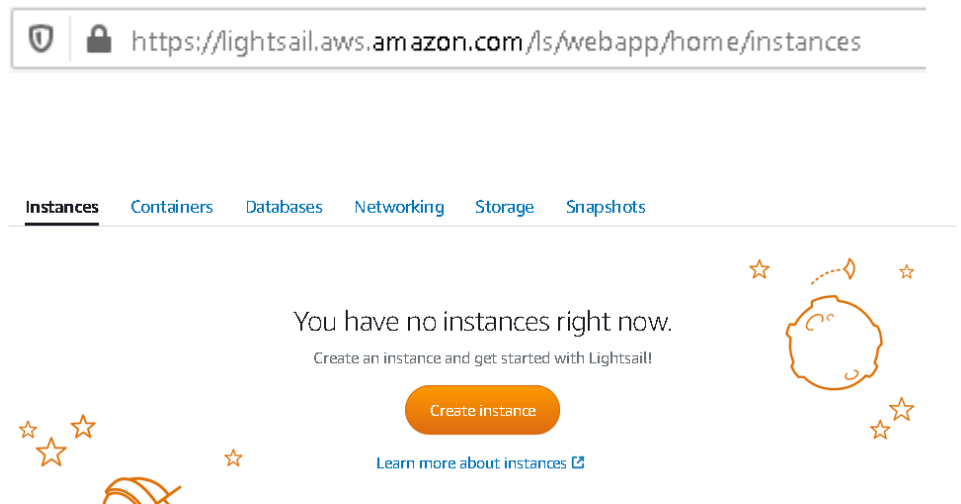
## Overview

1. [Create an AWS Lightsail instance](#)
2. [Create Static IP](#)
3. [Setup Firewall](#)
4. [Upload witness software](#)
5. [Install witness software](#)
6. [Configure witness software as a service](#)
7. [Add witness server to ExpressCluster](#)

## Step-by-step

### Step 1 - Create an AWS Lightsail instance

- Login to [AWS Lightsail Homepage](#)
- Click Create instance



- Chose the region for the instance

## Instance location ?



You are creating this instance in **Virginia, Zone A** (us-east-1a)

[Change AWS Region and Availability Zone](#)

- Chose Linux/UNIX platform

## Pick your instance image ?

Select a platform

 **Linux/Unix**  
26 blueprints

 **Microsoft Windows**  
4 blueprints

- Select the Node.js blueprint

 **WordPress**  
5.4.2

 **WordPress Multisite**  
5.4.2

 **LAMP (PHP 7)**  
7.4.7

 **Node.js**  
12.18.3

 **Joomla**  
3.9.19

 **Magento**  
2.3.5-1

 **MEAN**  
4.2.8-7

 **Drupal**  
9.0.1

 **GitLab CE**  
12.5.0

 **Redmine**  
4.1.1-2

 **Nginx**  
1.18.0-3

 **Ghost**  
3.28.0

 **Django**  
3.1.0

 **Plesk Hosting Stack on Ubuntu**  
18.0.28

 **cPanel & WHM for Linux**  
11.90.0.10

- Click the “Change SSH key pair” link
- Click “Create New”

You are using the **default** SSH key pair for connecting to your instance.

[Change SSH key pair](#)

## SSH key pair manager ?

Select, create, or upload the key pair you would like to use to SSH into your instance.

[Create New](#) + [Upload New](#) 

☒ **Default** ?

[Download](#) 

- Click Create

## SSH key pair region



You are creating this SSH key pair in **Virginia, all zones** (us-east-1).

[Learn more about AWS Regions and Availability Zones](#)

Create

Cancel

Create a new SSH key pair

- Choose a name for the key pair and click Generate key pair.

We can generate an SSH key pair for you.  
We will keep the public key, and you can download the private key for later use.

witness

Generate key pair

Cancel

- Click download key

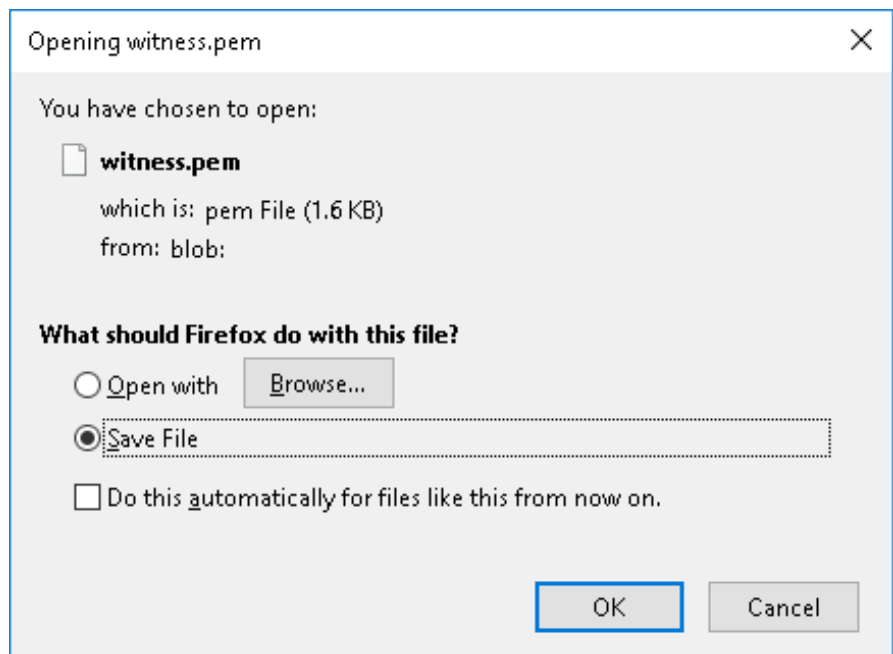
## Key pair created!

Your key pair has been successfully created. Download your private key now.

**You can only download this private key once.**

Download Key

- Save the private key file to somewhere convenient.



- Verify that the new key pair is selected instead of the Default.

## SSH key pair manager ?

Select, create, or upload the key pair you would like to use to SSH into your instance.

Create New + Upload New

☐ Default ?

Download

☒ witness

X

## Identify your instance

- Chose a name for the instance.

Your Lightsail resources must have unique names.

Node-js-2

x 1

- Click Create Instance

Create

- Wait for status to change from pending to running



Node-js-2

512 MB RAM, 1 vCPU, 20 GB SSD

Pending

52.87.188.246

Virginia, Zone A



Node-js-2

512 MB RAM, 1 vCPU, 20 GB SSD

Running

52.87.188.246

Virginia, Zone A

## Step 2 - Create static IP

- Click Networking

Instances

Containers

Databases

Networking

Storage

Snapshots

You have no instances right now.

Create an instance and get started with Lightsail!

Create instance

[Learn more about instances](#)

Create static IP

- Click Create static IP
- Select the instance to attach the IP to



WordPress  
5.4.2



WordPress  
Multisite  
5.4.2



LAMP (PHP 7)  
7.4.7



Node.js  
12.18.3



Joomla  
3.9.19



Magento  
2.3.5-1



MEAN  
4.2.8-7



Drupal  
9.0.1



GitLab CE  
12.5.0



Redmine  
4.1.1-2



Nginx  
1.18.0-3



Ghost  
3.28.0



Django  
3.1.0



Plesk Hosting  
Stack on  
Ubuntu  
18.0.28



cPanel & WHM  
for Linux  
11.90.0.10

- Choose a name to identify the static IP

## Identify your static IP

Your Lightsail resources must have unique names.

StaticIp-2

- Click create

Create

- Verify and take note of static IP information

### Public static IP address

This static IP is available for public connection worldwide.

23.21.9.100

### Attach to an instance

Attaching a static IP replaces that instance's dynamic IP address.



Node-js-2

512 MB RAM, 1 vCPU, 20 GB SSD  
Node.js

Detach ✕

## Step 3 - Setup Firewall

- From the AWS Lightsail home page, click on the toolbar radio on the right side of the instance and click manage
- Click Networking
- Under Firewall click Add rule



Node-js-2

512 MB RAM, 1 vCPU, 20 GB SSD

Running

Connect

Manage

Stop

Reboot

Delete

Connect

Storage

Metrics

Networking

Snapshots

Tags

History

Delete

## Firewall ?

Create rules to open ports to the internet, or to a specific IP address or range.

[Learn more about firewall rules](#)

+ Add rule

Application	Protocol	Port or range / Code	Restricted to	
SSH	TCP	22	Any IP address Lightsail browser SSH/RDP ?	

HTTP	TCP	80	Any IP address	 
HTTPS	TCP	443	Any IP address	 





- Set the port to 29009

Application:  Protocol:  Port or range:  ☐ Restrict to IP address

**NOTE:** You can click the “Restrict to IP address” box to add the public IP(s) of the clustered servers to strictly limit connections on this port to the clustered servers.

- Click create
- Verify the new rule is listed

Create 

Application	Protocol	Port or range / Code	Restricted to	
SSH	TCP	22	Any IP address Lightsail browser SSH/RDP 	 
HTTP	TCP	80	Any IP address	 
HTTPS	TCP	443	Any IP address	 
Custom	TCP	29009	Any IP address	 

## Step 4 - Upload witness software

- Launch WinSCP
- Fill out the session settings as shown.  
NOTE: Be sure the username is set to “bitnami”



Login

New Site  
ec2-user@3.89.154.125

Session

File protocol:

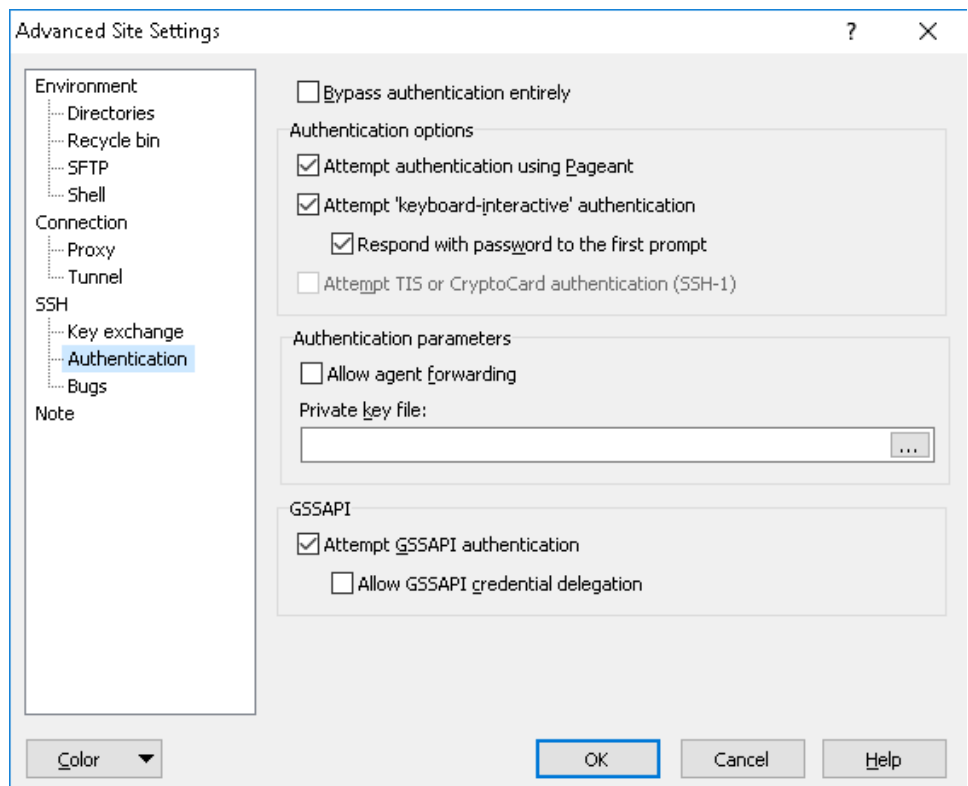
Host name:  Port number:

User name:  Password:

Tools

Advanced...

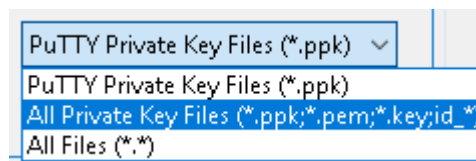
- Click the Advanced button
- Highlight Authentication on the left



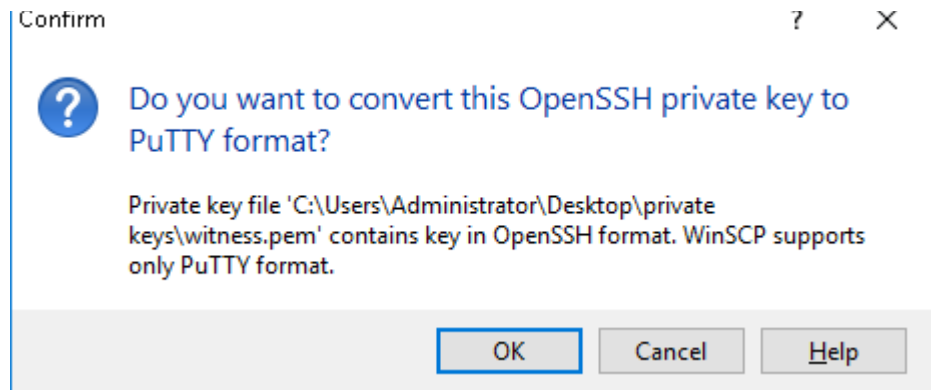
- Under Authentication parameters, click the “...” button to select a private key file



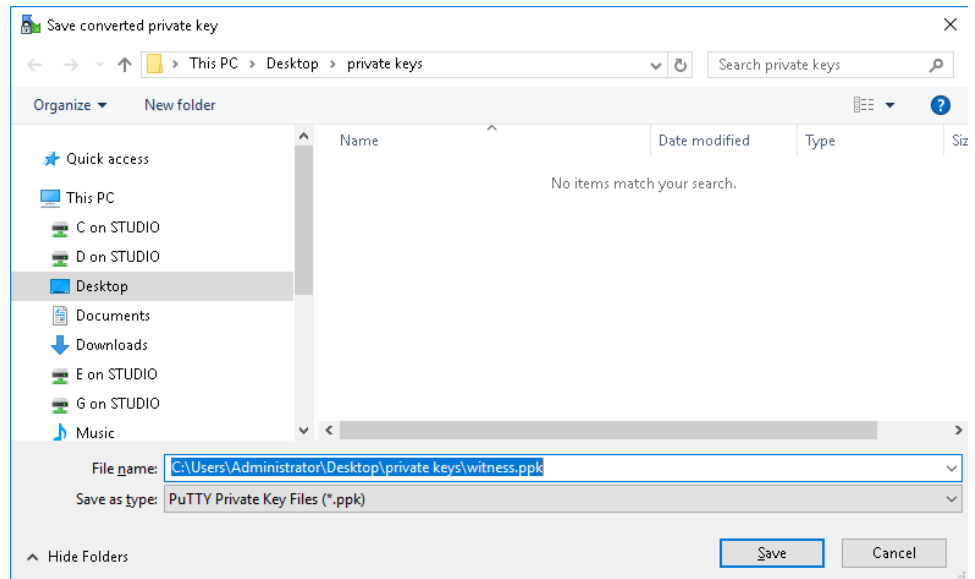
- Click the dropdown on the bottom right, select “All private key files”, and open the private key that was downloaded previously from AWS



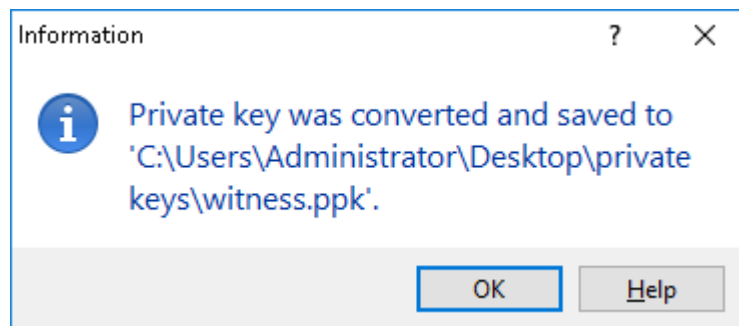
- When prompted to convert the key to putty format, click ok



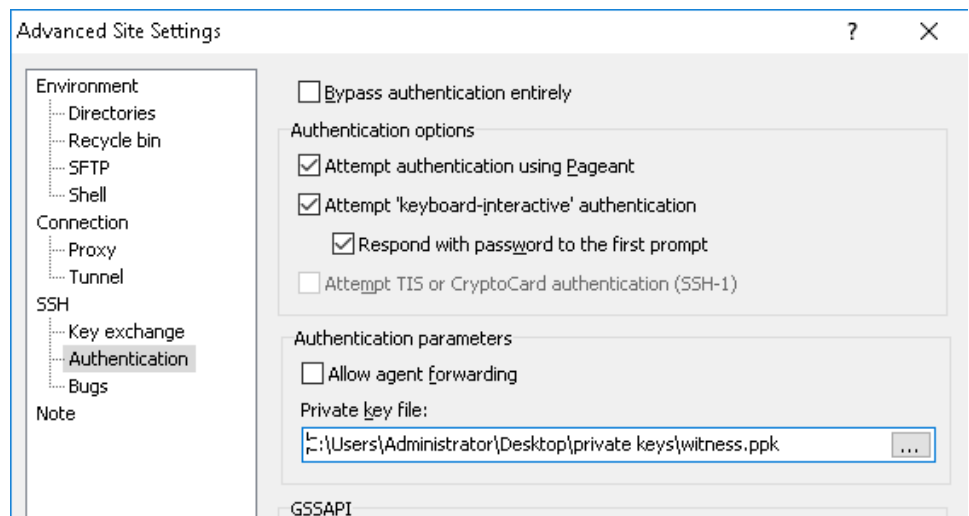
- Save the newly converted key



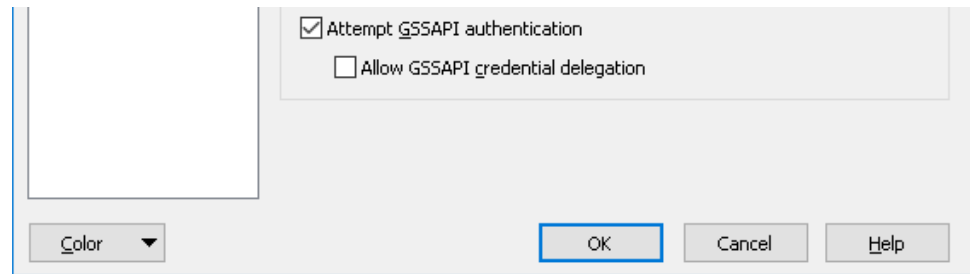
- Click ok



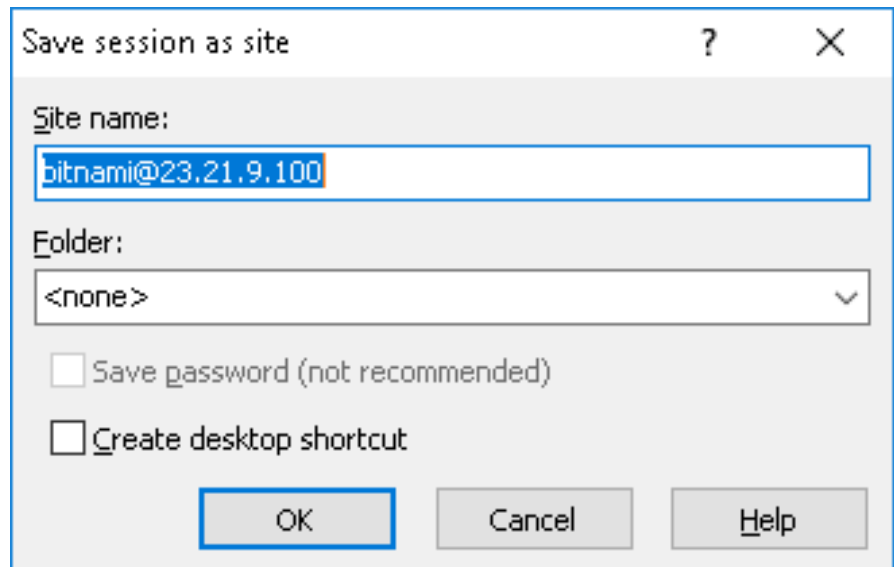
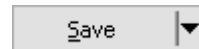
- Click ok again



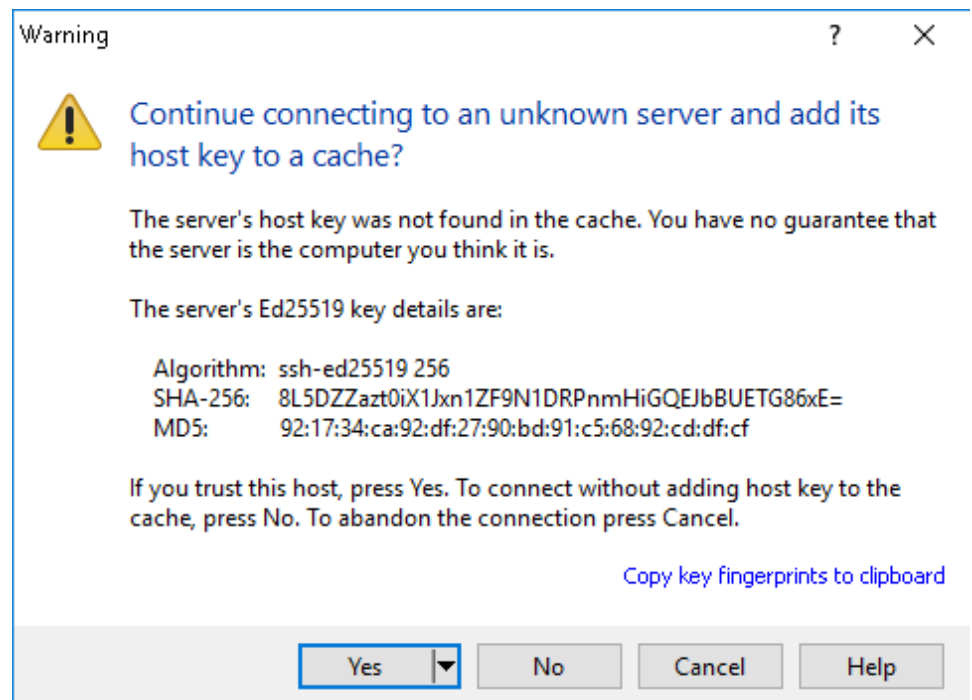





- Click Save
- Choose a session name and click ok



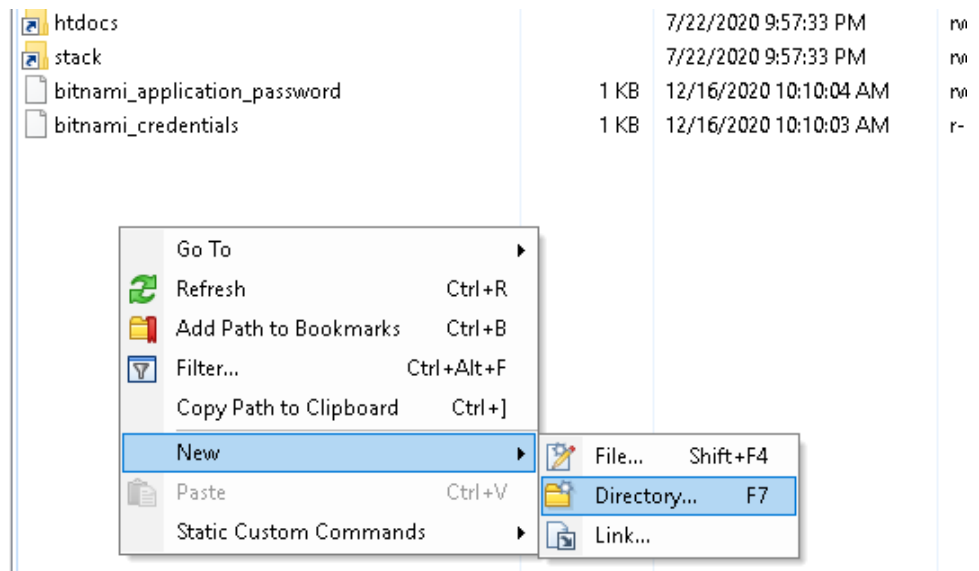
- Click Login
- Click yes to trust the host



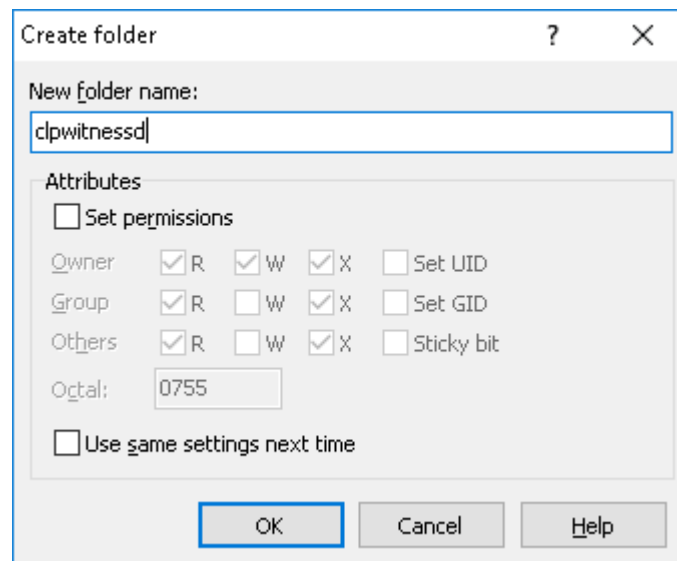
- On the host side file browser right

/home/bitnami/			
Name	Size	Changed	Ri
		7/22/2020 9:51:34 PM	rw

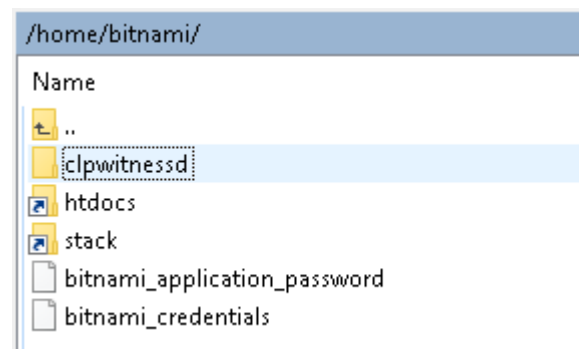
click and select  
New->Directory



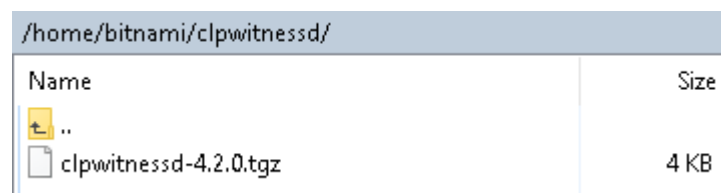
- Name the directly  
“clpwitnessd” and  
click ok



- Double click the  
clpwitnessd folder  
to change directory



- Upload  
clpwitnessd-  
<version>.tgz to  
the “/home/bitnami  
/clpwitnessd” folder



## Step 5 - Install witness software

- From the AWS Lightsail home page, click on the toolbar radio on the right side of the instance and click manage



- Click Connect using SSH

### Connect securely using your browser [?](#)

You can still use your own compatible ssh client with your device or software to connect to your instance. [Learn how to connect using your own SSH client](#)

Connect using SSH

**Note:** Alternatively, you can configure putty or any other ssh client to connect, using the private key created in previous steps.

- Install the witness software by executing the following command

```
$ cd ~/clpwitnessd
$ sudo install --global clpwitnessd-<version>.tgz
```

- If successful, the following should be displayed

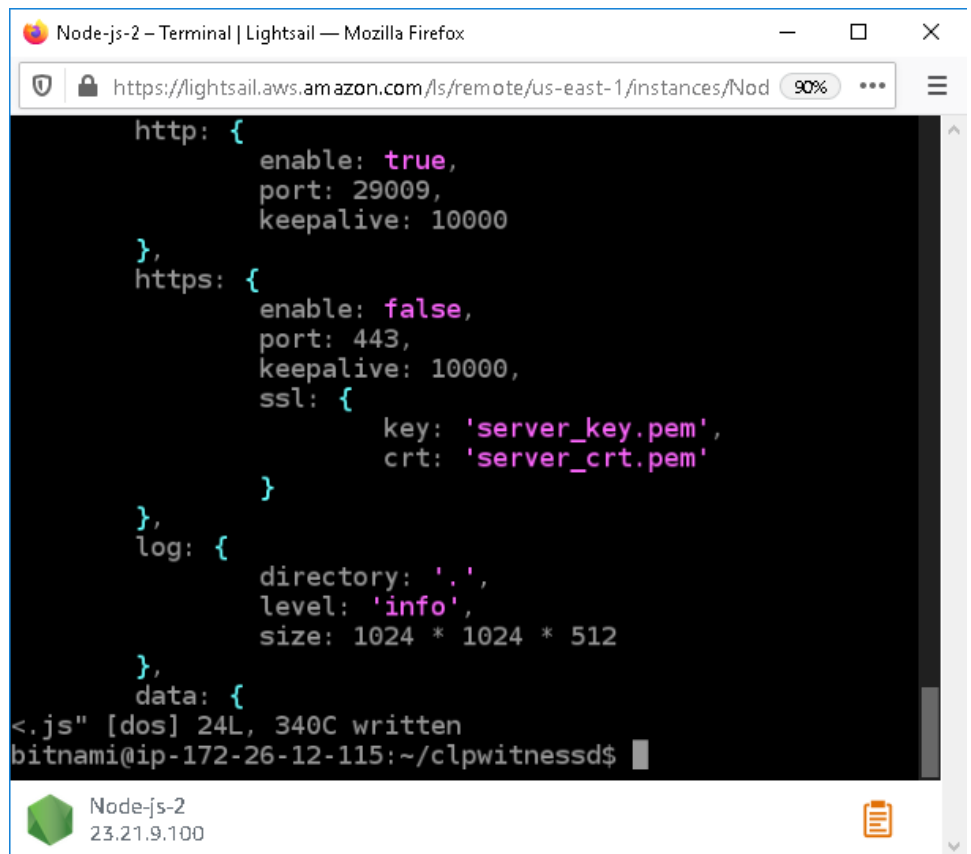
```
Node.js-2 - Terminal | Lightsail — Mozilla Firefox
https://lightsail.aws.amazon.com/lightsail/remote/us-east-1/instances/Nod 90% ...
bitnami@ip-172-26-12-115:~/clpwitnessd$ sudo npm install -g clpwitnessd-4.2.0.tgz
/opt/bitnami/node/bin/clpwitnessd -> /opt/bitnami/node/lib/node_modules/clpwitnessd/clpwitnessd.js
+ clpwitnessd@4.2.0
updated 1 package in 0.123s
bitnami@ip-172-26-12-115:~/clpwitnessd$
```



- Edit the witness config file

```
$ sudo vi /opt/bitnami/node/lib/node_modules/clpwitnessd/clpwitnessd.conf.js$ sudo vi /opt/bitnami/node/lib/node_modules/clpwitnessd/clpwitnessd.conf.js
```

- Under the http section of the file, change the port from 80 to 29009 and save



## Step 6 - Configure witness software as a service

- Create and edit the clpwitnessd.service file

```
$ sudo vi /etc/systemd/system/clpwitnessd.service
```

- Copy or paste the following text into the clpwitnessd.service file and save

```
[Unit]
Description=ECX AWS Witness
After=syslog.target network.target
[Service]
Type=simple
ExecStart=/opt/bitnami/node/bin/clpwitnessd
WorkingDirectory=/opt/bitnami/node/lib
/node_modules/clpwitnessd
KillMode=process
Restart=always
[Install]
WantedBy=multi-user.target
```

- Edit clpwitnessd script.

```
$ sudo vi /opt/bitnami/node/bin/clpwitnessd
```

- Edit the shebang in the script to reflect the proper location of the “node” executable.

Find the line that shows..

```
#!/usr/bin/env node
```

and change to the following...

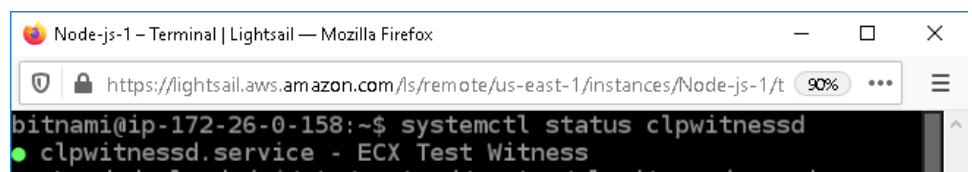
```
#!/usr/bin/env /opt/bitnami/node/bin/node
```

- Enable the clpwitnessd service
- Start the clpwitnessd service
- Verify the clpwitnessd service is running properly.

```
$ sudo systemctl enable clpwitnessd
```

```
$ sudo systemctl start clpwitnessd
```

```
$ sudo systemctl status clpwitnessd
```



The screenshot shows a terminal window titled "Node.js-1 - Terminal | Lightsail - Mozilla Firefox". The address bar shows the URL "https://lightsail.aws.amazon.com/lightsail/remote/us-east-1/instances/Node.js-1/t". The terminal output shows the command "systemctl status clpwitnessd" and the output "clpwitnessd.service - ECX Test Witness". The service is shown as "loaded" and "active (running)".

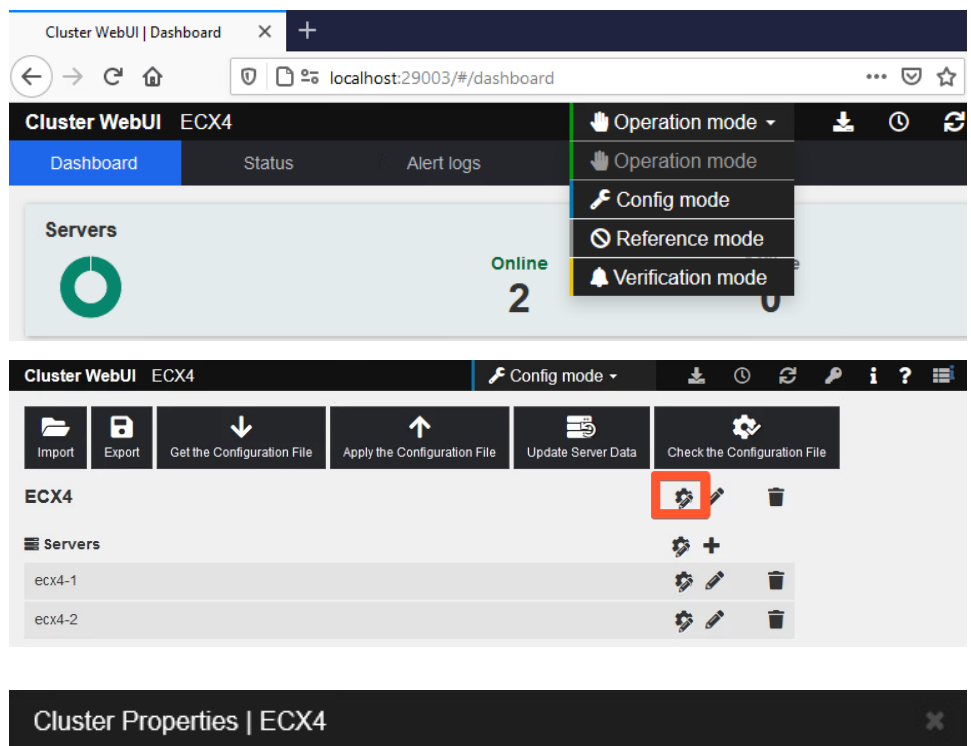
```
Loaded: loaded (/etc/systemd/system/clpwitnessd.service; ena
Active: active (running) since Wed 2020-12-16 19:45:07 UTC;
Main PID: 504 (node)
Tasks: 11 (limit: 558)
Memory: 32.3M
CGroup: /system.slice/clpwitnessd.service
└─504 /opt/bitnami/node/bin/node /opt/bitnami/node/b

Dec 16 19:45:07 ip-172-26-0-158 systemd[1]: Started ECX Test Wi
lines 1-10/10 (END)
```

Node.js-1  
34.238.73.73

## Step 7 - Add witness server to ExpressCluster

- Launch the ExpressCluster web manager
- Switch to config mode
- Click the cluster properties button
- Click the Interconnect tab



[Disk](#)
[Mirror Disk](#)
[Account](#)
[RIP\(Legacy\)](#)
[Migration](#)
[Extension](#)

**Cluster Name** ECX4  
**Comment** Demo Cluster  
**Language** English

- Click the Add button

Cluster Properties | ECX4

[Info](#)
[Interconnect](#)
[NP Resolution](#)
[Timeout](#)
[Port No.](#)
[Monitor](#)
[Recovery](#)

[Alert Service](#)
[WebManager](#)
[API](#)
[Encryption](#)
[Alert Log](#)
[Delay Warning](#)

[Disk](#)
[Mirror Disk](#)
[Account](#)
[RIP\(Legacy\)](#)
[Migration](#)
[Extension](#)

Heartbeat I/F Priority List

Priority	Type	MDC	ecx4-1	ecx4-2
1	Kernel Mode	Do Not Use	192.168.1.101	192.168.1.102
2	Kernel Mode	mdc1	10.10.10.101	10.10.10.102

☒

☐ Broadcast
 ☒ Unicast

- Under the Type dropdown for the newly added I/F, select Witness

Priority	Type	MDC	ecx4-1	ecx4-2
1	Kernel Mode	Do Not Use	192.168.1.101	192.168.1.102
2	Kernel Mode	mdc1	10.10.10.101	10.10.10.102
3	Kernel Mode	Do Not Use		

☒

☐ Broadcast
 ☒ Unicast

- Make sure the new I/F is selected and click the Properties button

Cluster Properties | ECX4

[Info](#)
[Interconnect](#)
[NP Resolution](#)
[Timeout](#)
[Port No.](#)
[Monitor](#)

[Encryption](#)
[Alert Log](#)
[Delay Warning](#)
[Disk](#)
[Mirror Disk](#)
[Account](#)

Properties

Add

Remove

Heartbeat I/F Priority List

Priority	Type	MDC	ecx4-1
1	Kernel Mode	Do Not Use	192.168.1.10
2	Kernel Mode	mdc1	10.10.10.101
3	Witness	Do Not Use	Use

↑

↓

**Set up the target host in Witness Heartbeat properties.**

**Server Down Notification** ☒

☐ Broadcast
 ☒ Unicast

- Change the Target Host to the static IP associated with the AWS instance, set the Service Port to 29009 and click OK

Witness HeartBeat Properties

Target Host\*

23.21.9.100

Service Port\*

29009

Use SSL

☐

Use Proxy

☐

HTTP Timeout\*

10

sec

Initialize

OK

Cancel

- Click OK again
- Click Apply the Configuration File
- Click OK

OK

Import

Export

Get the Configuration File

Apply the Configuration File

Update Server Data

Check the Configuration File

Cluster WebUI

Apply the changes.  
To apply the changes, the following operations must be performed.

Suspend the cluster  
Restart Information Base service  
Restart WebManager service



Do you want to perform the operations?

OK

Cancel

- If you see the following messages, click OK

### Cluster WebUI

Changes applied successfully.  
Some services have been stopped in order to apply the changes.  
Use the following steps to resume the stopped services.

Resume the cluster

Execute now ?

OK

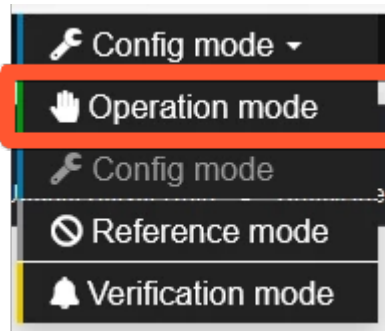
Cancel

### Cluster WebUI

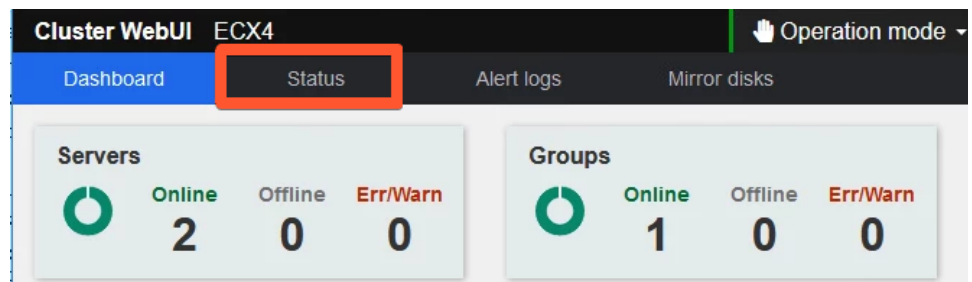
Restart Information Base service.  
Restart WebManager service.

OK

- Switch back to Operation Mode



- Click on Status



- Click the Server down-arrow to expand view



- Verify witnesshb1 and httpnp1 are Normal status for both servers

Server		Online	Online
▼ lankhb1		Normal	Normal
▼ lankhb2		Normal	Normal
▼ witnesshb1		Normal	Normal
▼ httpnp1		Normal	Normal

**i** NOTE: The HTTPnp resource is added automatically when the witness heartbeat is added to the cluster config. This can be removed later if desired without affecting the witness heartbeat.

**Related articles**