Cluster Deployment Guide

Version 2.2.2

Contents

Copyright Notice

Document Revision History

Friday, December 7th, 2018

- Initial release of documentation.

OVA Download

The latest OVA file is available as a secure download hosted on Amazon S3.

Your professional services representative will provide you with a secure link to download the file when it becomes available.

OVA Deployment

Preparations

To set up Band, you must have:

- Band OVA
- Supported virtual infrastructure
- PostgreSQL database credentials
- Nginx compatible SSL certificate and SSL certificate key

OVA Deployment

Network

Port Usage Outside of Cluster Group

Protocol	Port	Direction	Purpose
HTTPS	443	Inbound	Band API
HTTPS	4737	Inbound	IPFS Websockets
SSH	22	Inbound/Outbound	Cluster administration

Port Usage Inside of Cluster Group

Protocol	Port	Direction	Purpose
TCP	13000-14000	Inbound/Outbound	Cluster messaging
HTTPS	443	Inbound/Outbound	Band API
SSH	22	Inbound/Outbound	Cluster administration
TCP	5432*	Outbound	Database

* Subject to change depending on customer's preferred storage implementation.

OVA Deployment

System Requirements

Supported Platforms

VMware ESXI 5.5 and later are supported.

Cluster Size

The recommended size of a Band cluster is 2 nodes.

Virtual Machine Configuration

The requirements for a Band cluster node are:

CPU: 3 GHz dual core or 4 virtual processors
RAM: 8 GB
STORAGE: 80GB

Browsers

The Band interface is supported on the latest versions of Firefox, Internet Explorer, Edge, Chrome, and Safari.

Band OVA Deployment

Deploying the OVA

Deploy the OVA on your platform as you would any other OVA. Refer to your platform's documentation for instructions on deploying OVA files.

Cluster Setup

Clusters are headless and all nodes are functionally identical.

Individual Node DNS Entries

Individual nodes do not require distinct DNS entries but can be assigned one for administrative convenience.

Load Balancing

Nodes do not require session affinity and utilize long-lived websocket connections.

The cluster can operate in two load balancing configurations:

Round-Robin DNS

All node IP addresses are assigned to a single DNS entry.

Hardware (Websocket Enabled)

Nodes can be used with hardware load balancers such as those available from Cisco or F5 for fault-tolerance. Hardware load balancers **must be configured for use with websockets**. Refer to your load balancer's documentation for instructions on enabling websockets.

SSL Certificates

All cluster nodes share a single SSL certificate and certificate key to communicate with external services.

The SSL certificate and certificate key should be Nginx compatible. See - http://nginx.org/en/docs/http/configuring_https_servers.html - for more information.

Node Setup

Network Setup (DHCP)

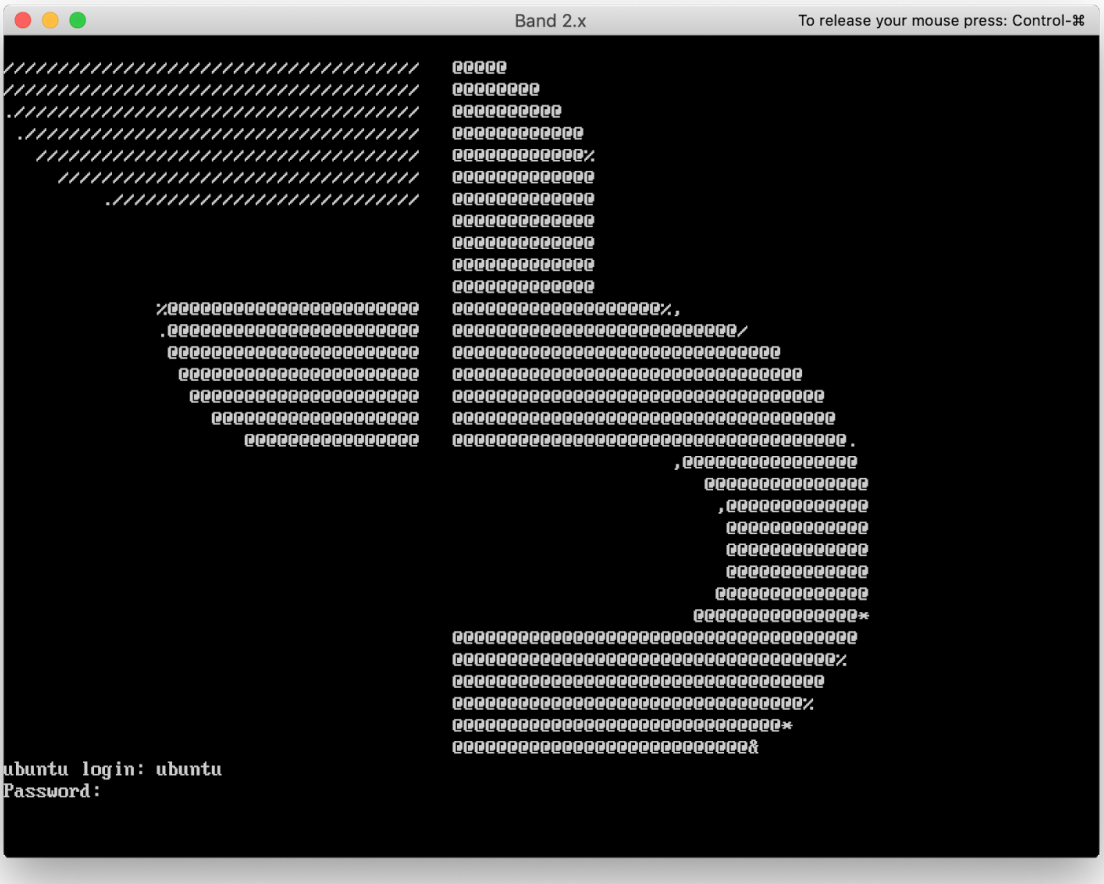
By default, nodes use dynamic host configuration protocol (DHCP) on network device ens32. No additional network setup is required on DHCP systems.

Network Setup (Static IP)

For systems with statically allocated IP addresses:

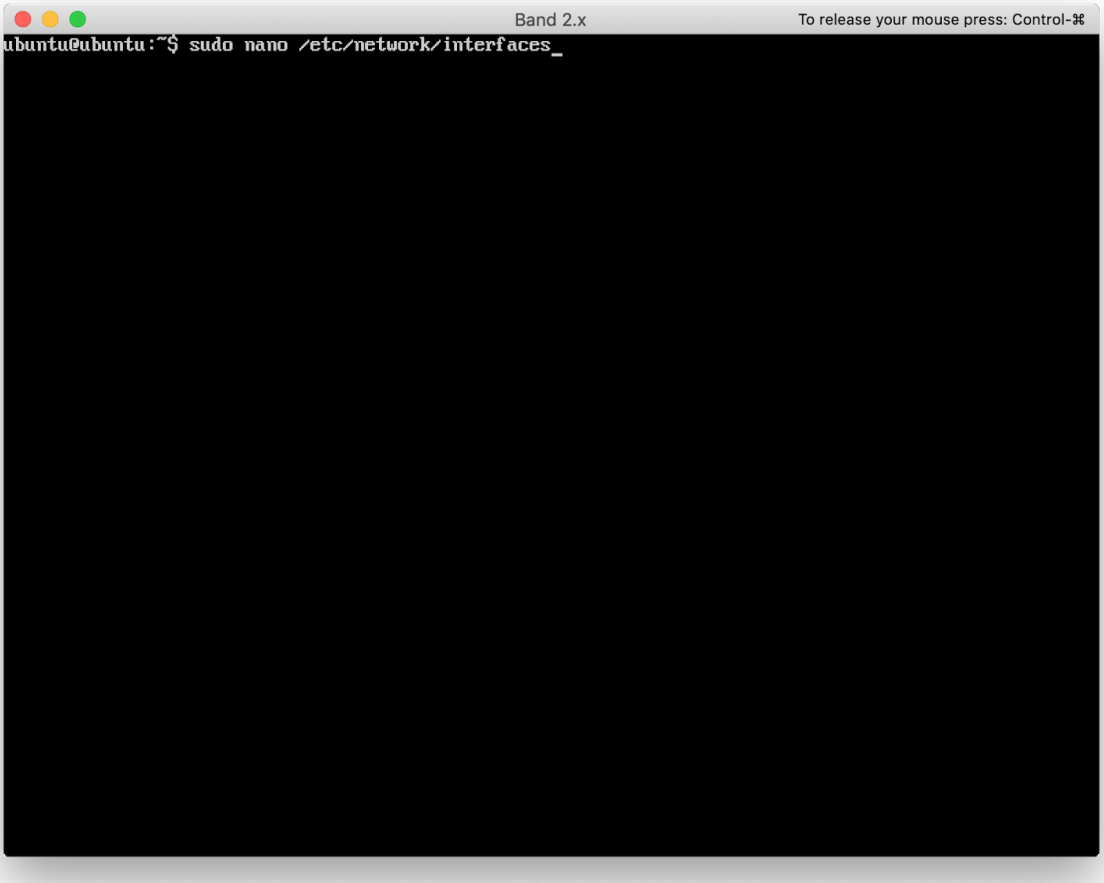
1. Access the virtual machine terminal.
2. At the login prompt, enter:

```
username: ubuntu
password: ubuntu
```

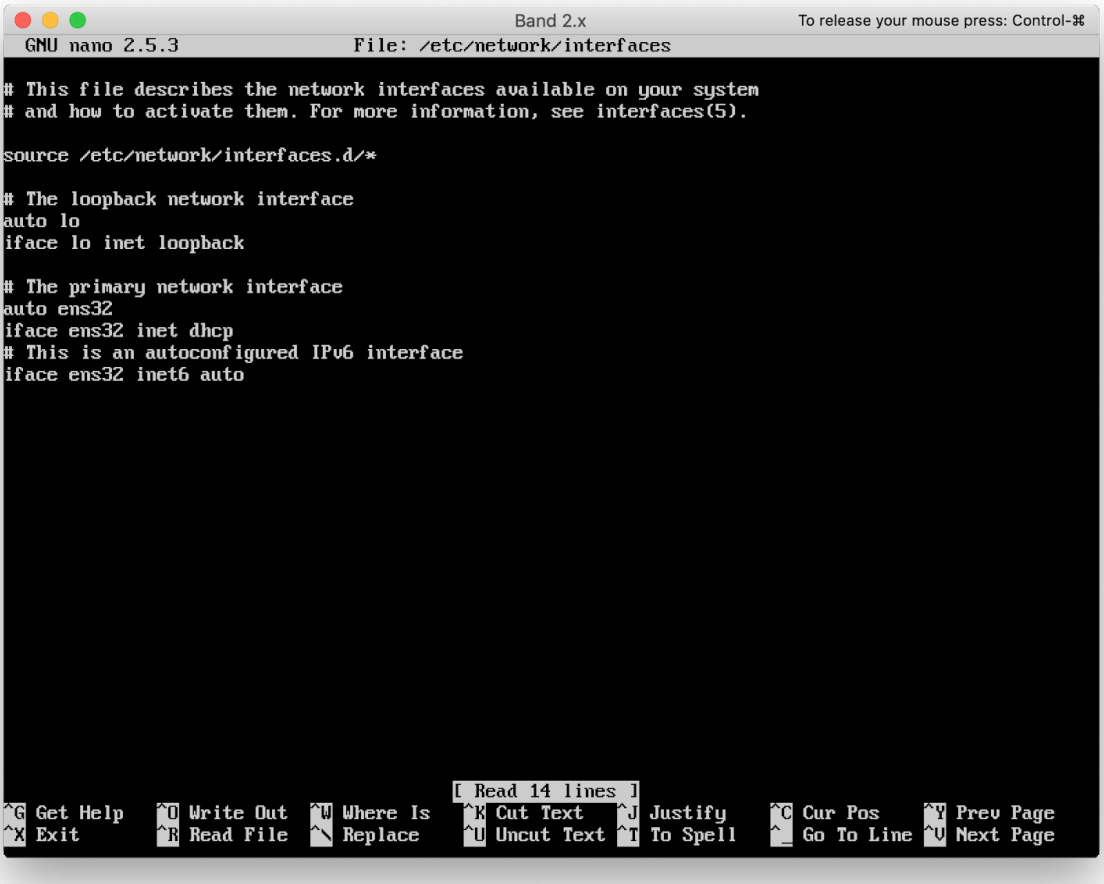


3. Open the network configuration file for editing:

```
sudo nano /etc/network/interfaces
```



4. Review and modify the settings as needed.

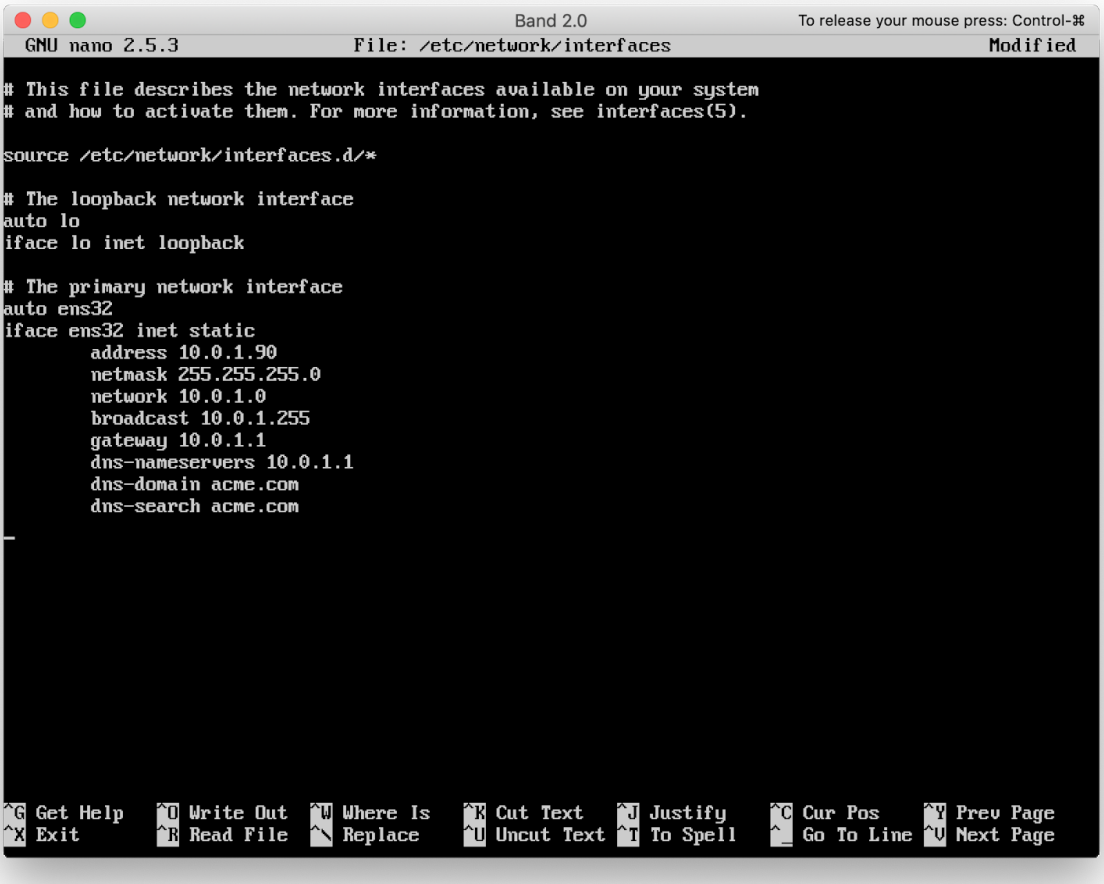


   - The file will look similar to:

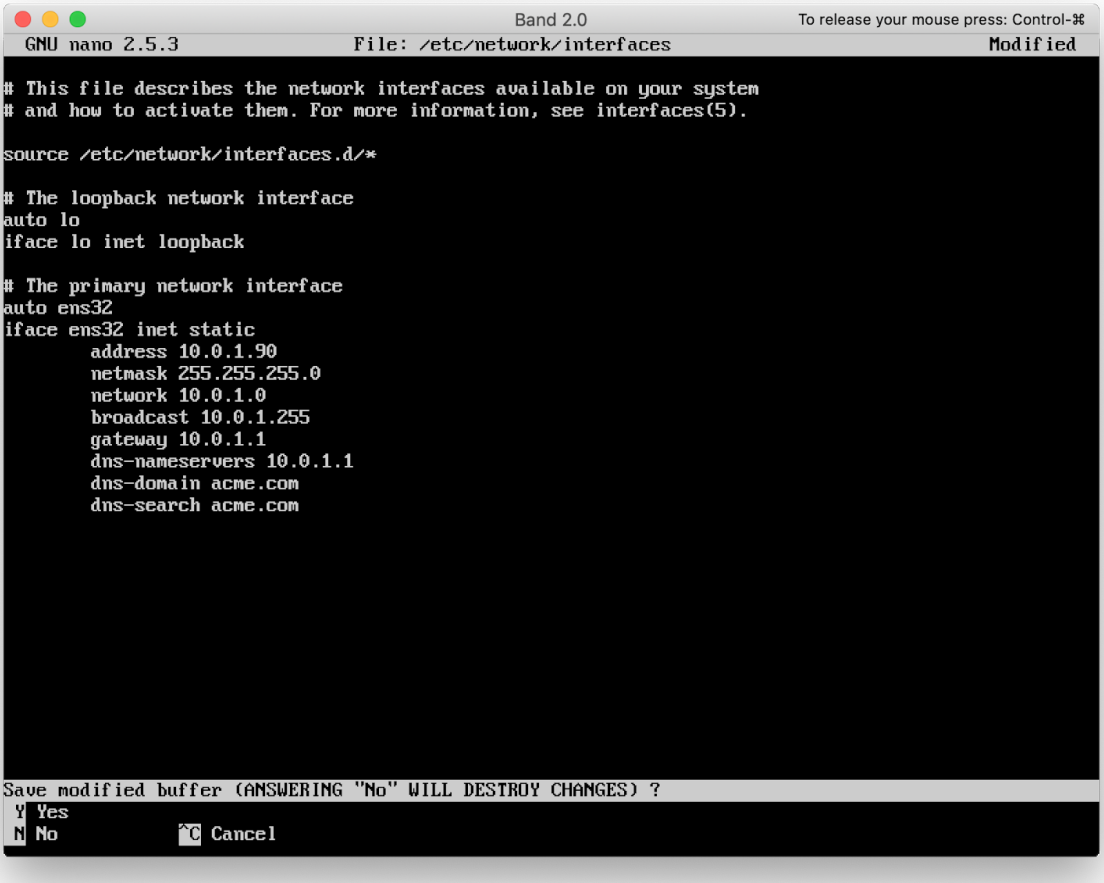
```
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto ens32
iface ens32 inet dhcp
# This is an autoconfigured IPv6 interface
iface ens32 inet6 auto
```

- Your changes will most likely look similar to:

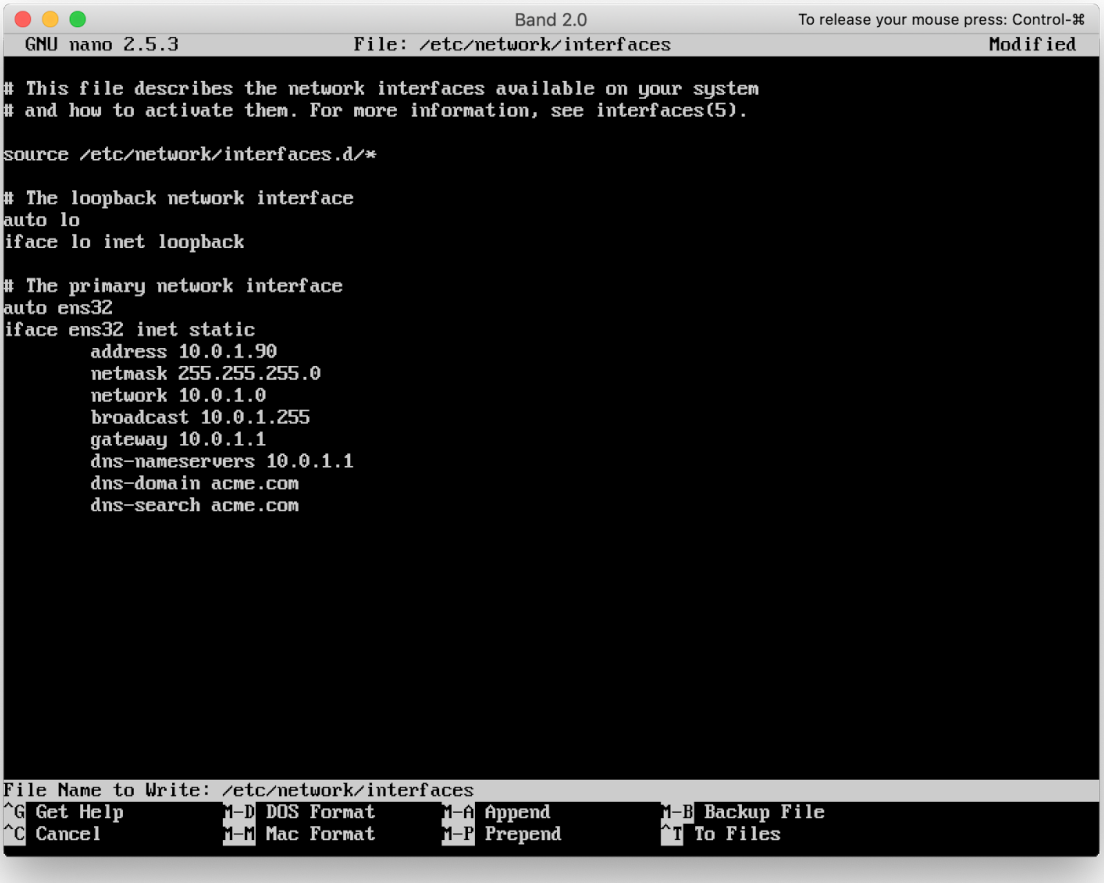
```
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto ens32
iface ens32 inet static
    address 10.0.1.90
    netmask 255.255.255.0
    network 10.0.1.0
    broadcast 10.0.1.255
    gateway 10.0.1.1
    dns-nameservers 10.0.1.1
    dns-domain acme.com
    dns-search acme.com
```



5. When your modifications are completed press **CTRL-X** to exit.
6. Press the **Y** key to save your changes.



7. Press **ENTER** to save the file.



8. Restart the networking stack:

   ```
   sudo systemctl restart networking
   ```

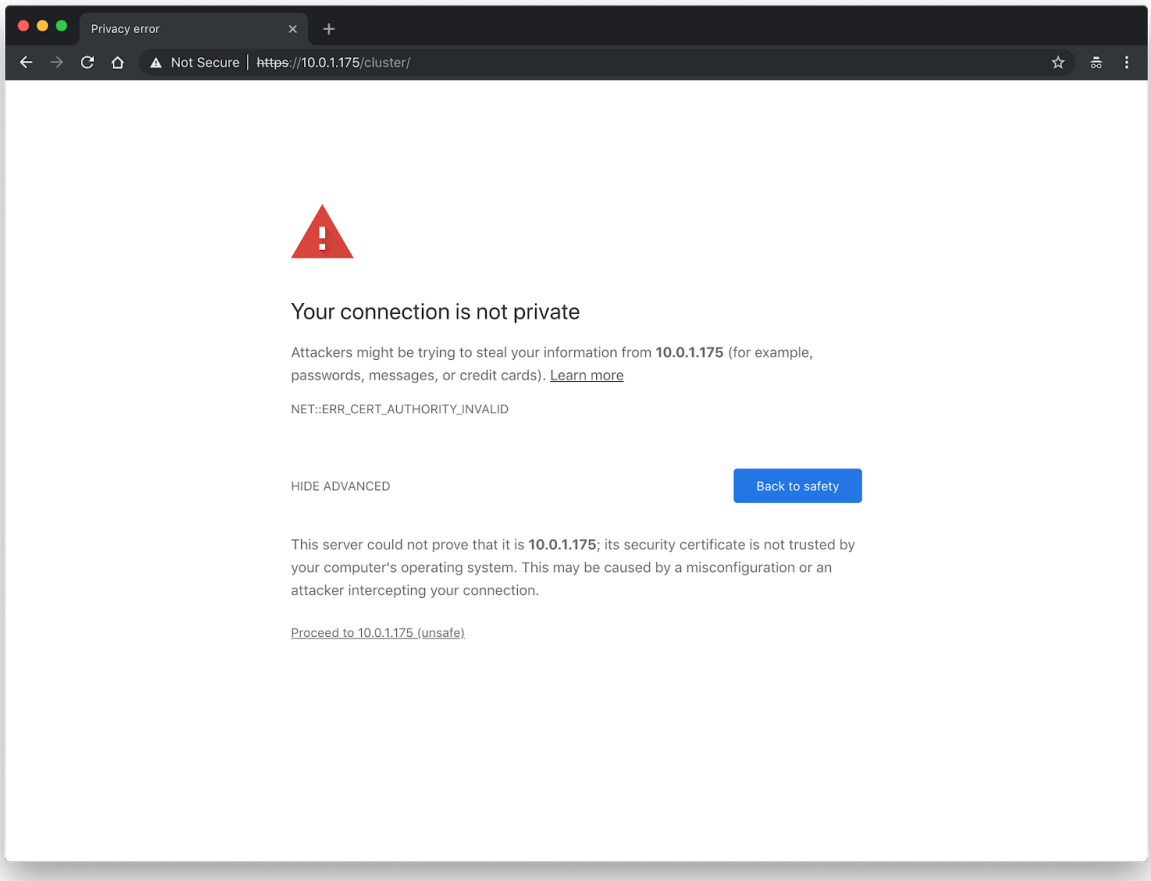
9. Reboot the virtual machine:

   ```
   sudo reboot
   ```

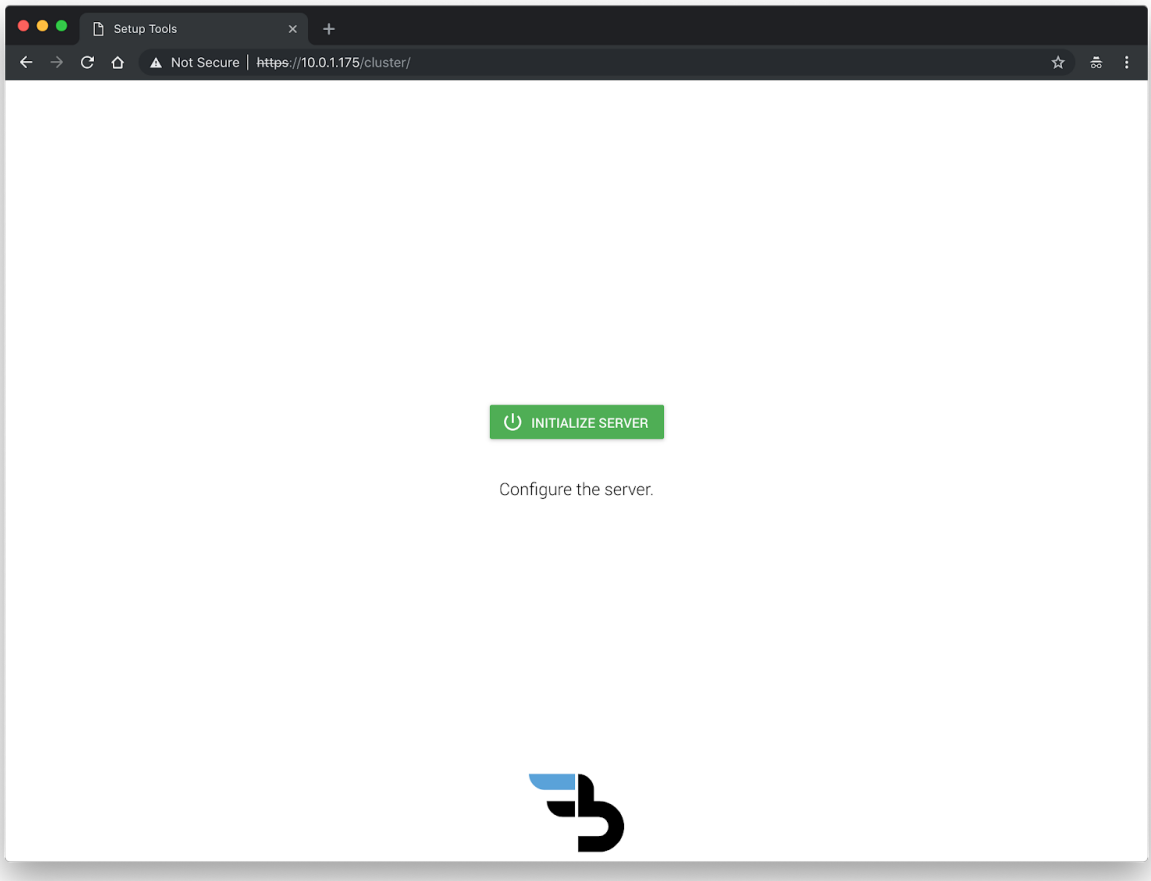
10. After the system restarts, confirm that it was configured successfully.
    - Ping the configured IP address:
      ping [configured IP address]
    - Access **https://[configured IP address]/cluster** in a web browser and check for the cluster setup screen.

Initialize Cluster

Visit the HTTPS **/cluster** path of the first node. If the node IP were **10.0.1.175,** the address would be **https://10.0.1.175/cluster**. Proceed through the SSL certificate warnings.

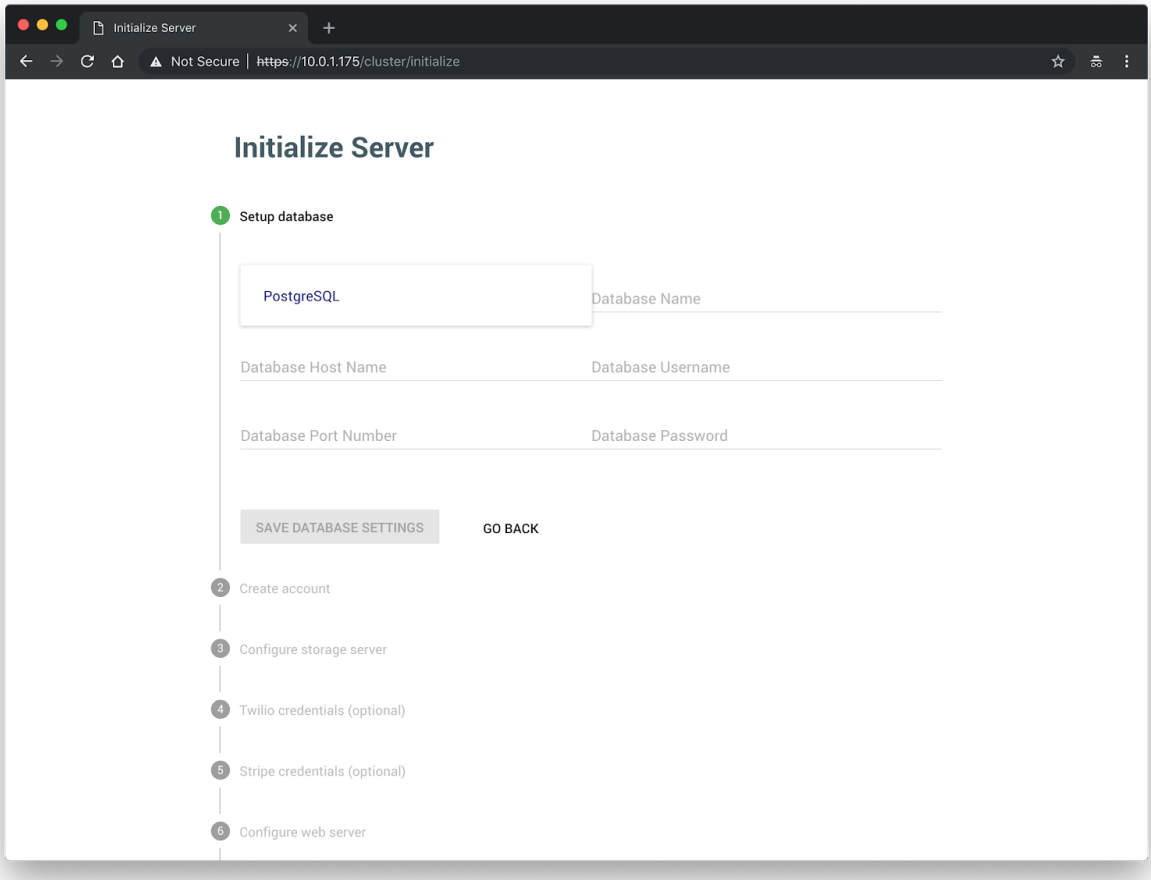


From the landing page, click on **Initialize Server** button.

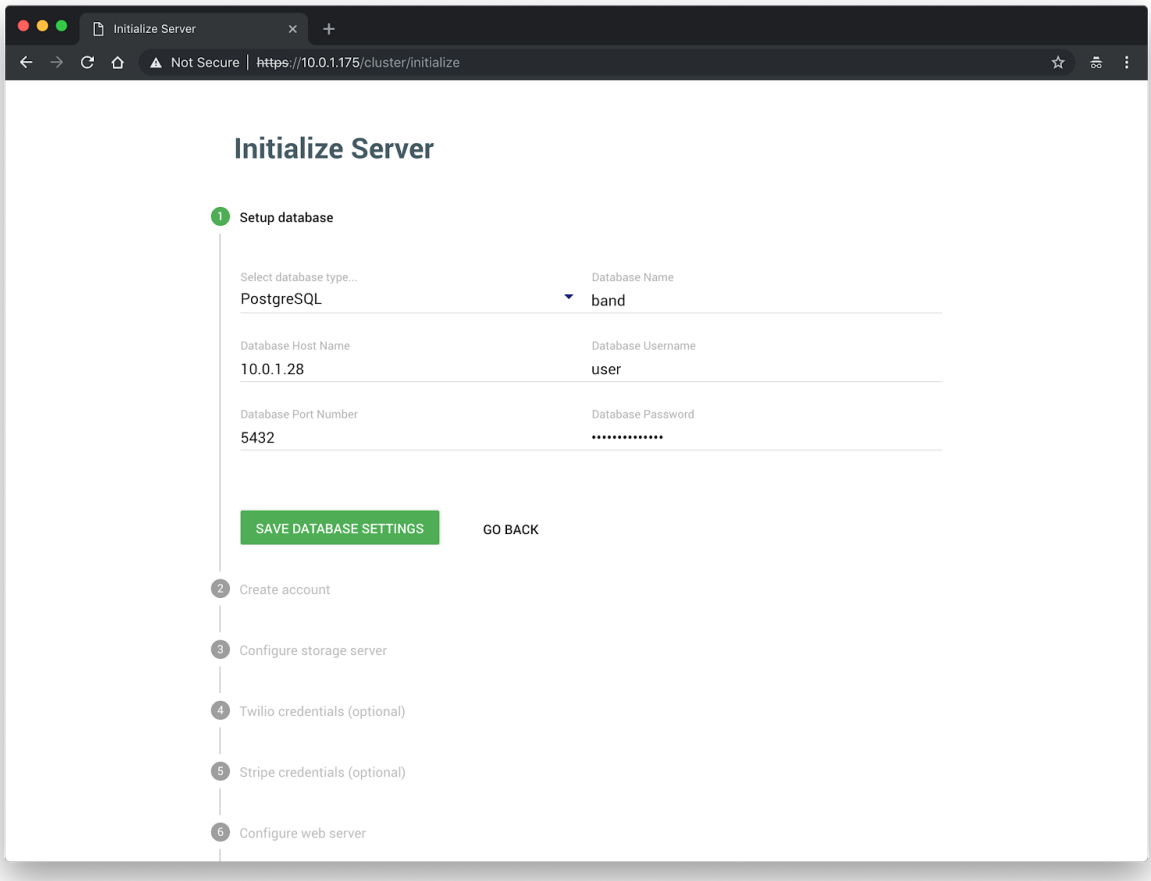


From **Initialize Server: Setup database**, enter the credentials of a previously set up Postgres database.

Click on the **Select database type** dropdown menu and choose PostgreSQL.

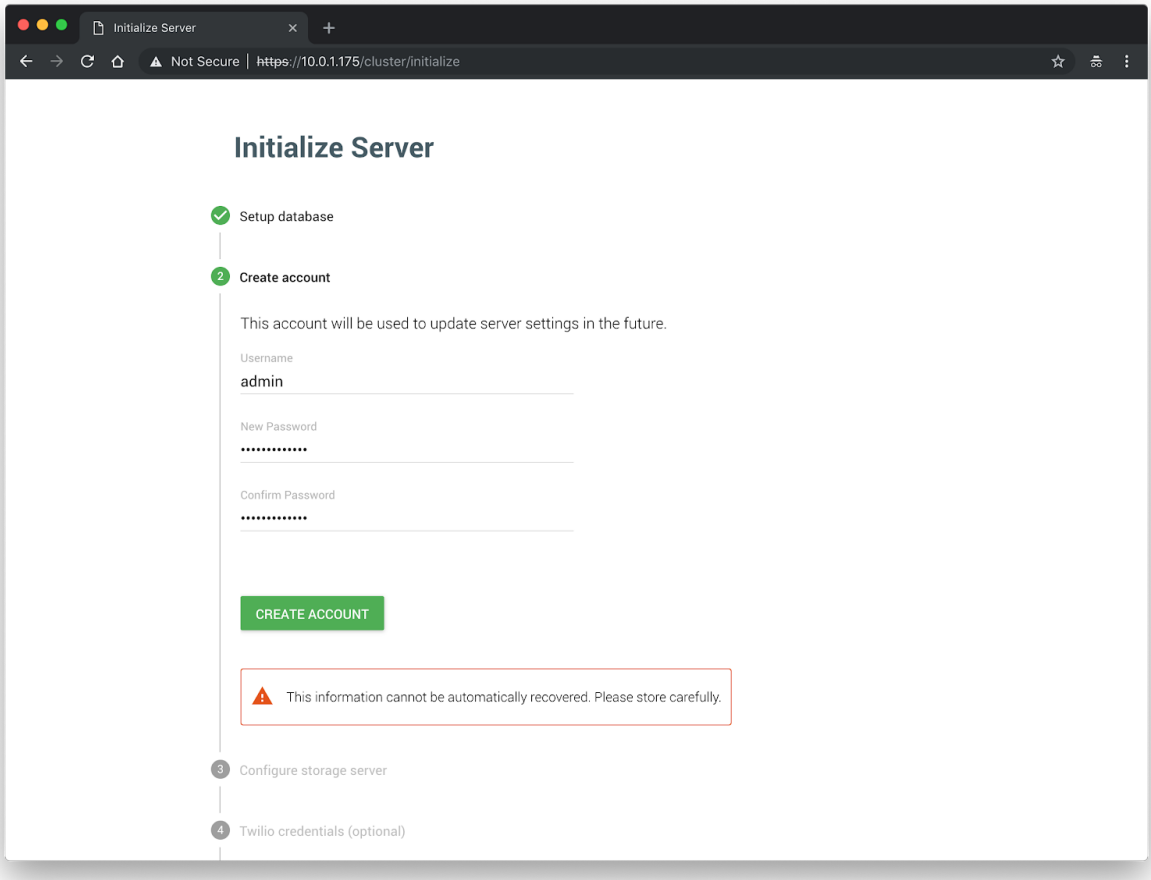


Enter the database credentials and click on **Save Database Settings** button.

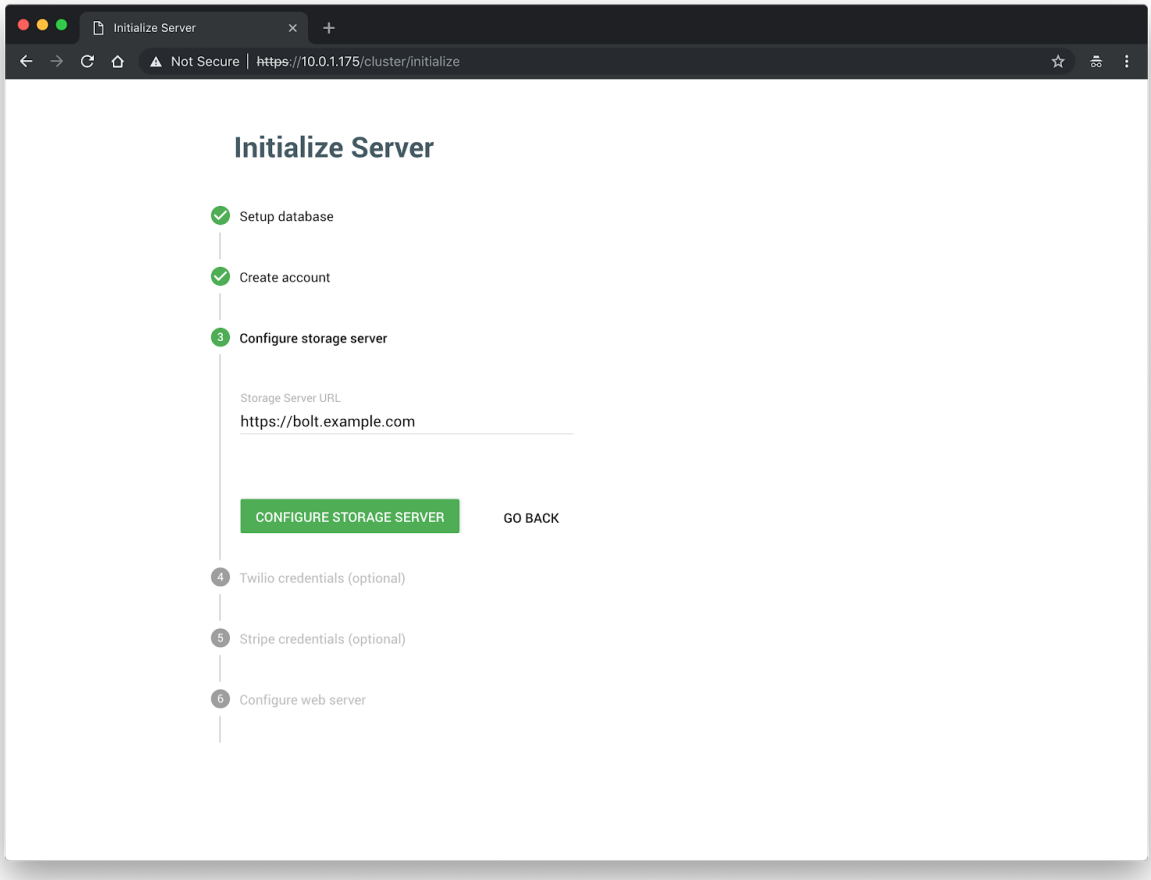


From **Initialize Server: Create account**, enter a username and enter the same password under both **New Password** and **Confirm Password** fields and click on **Create Account** in order to create cluster administrative account. Make a note of these credentials since they are required to access the cluster administration page.

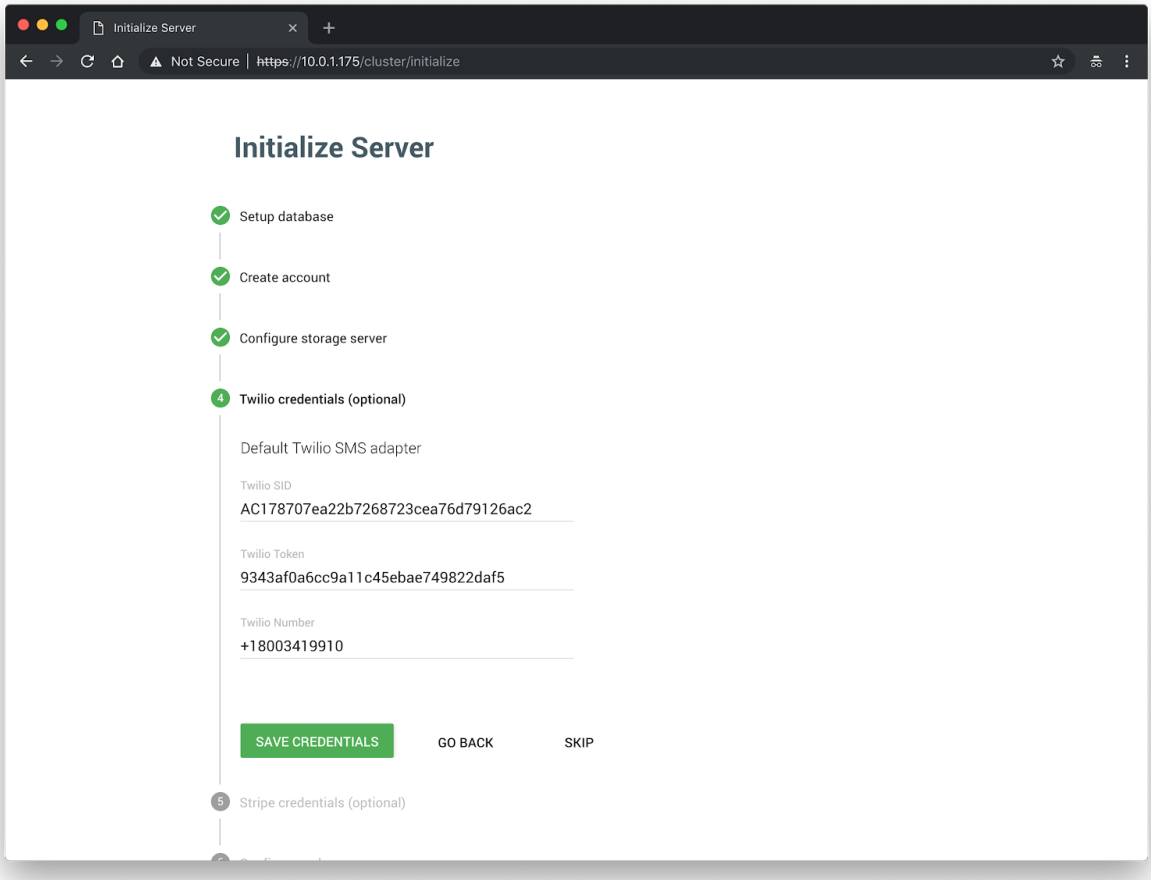
NOTE Credentials entered in this step cannot be recovered automatically.



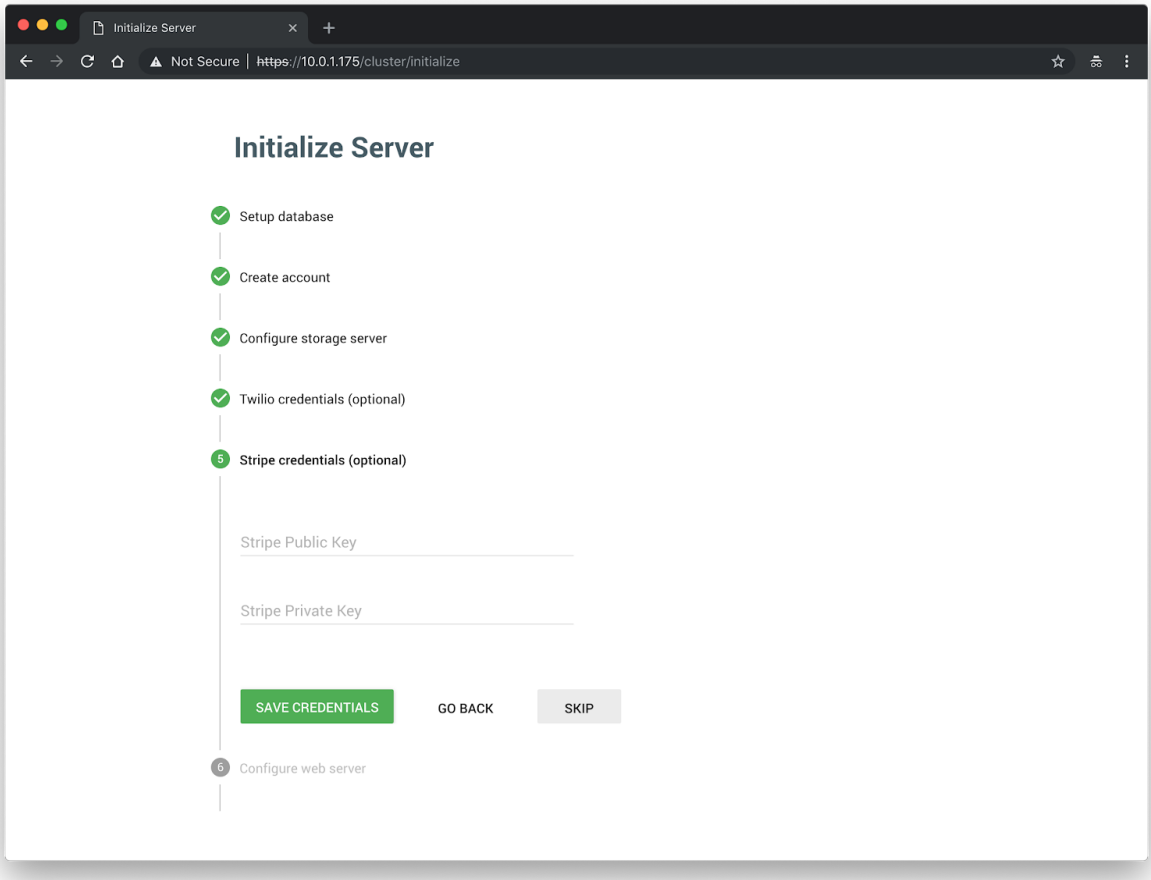
From **Initialize Server: Configure storage server**, enter the URL of the bolt cluster and click on **Configure Storage Server** button. For example, if bolt is hosted on **bolt.example.com** then the URL would be **https://bolt.example.com.**



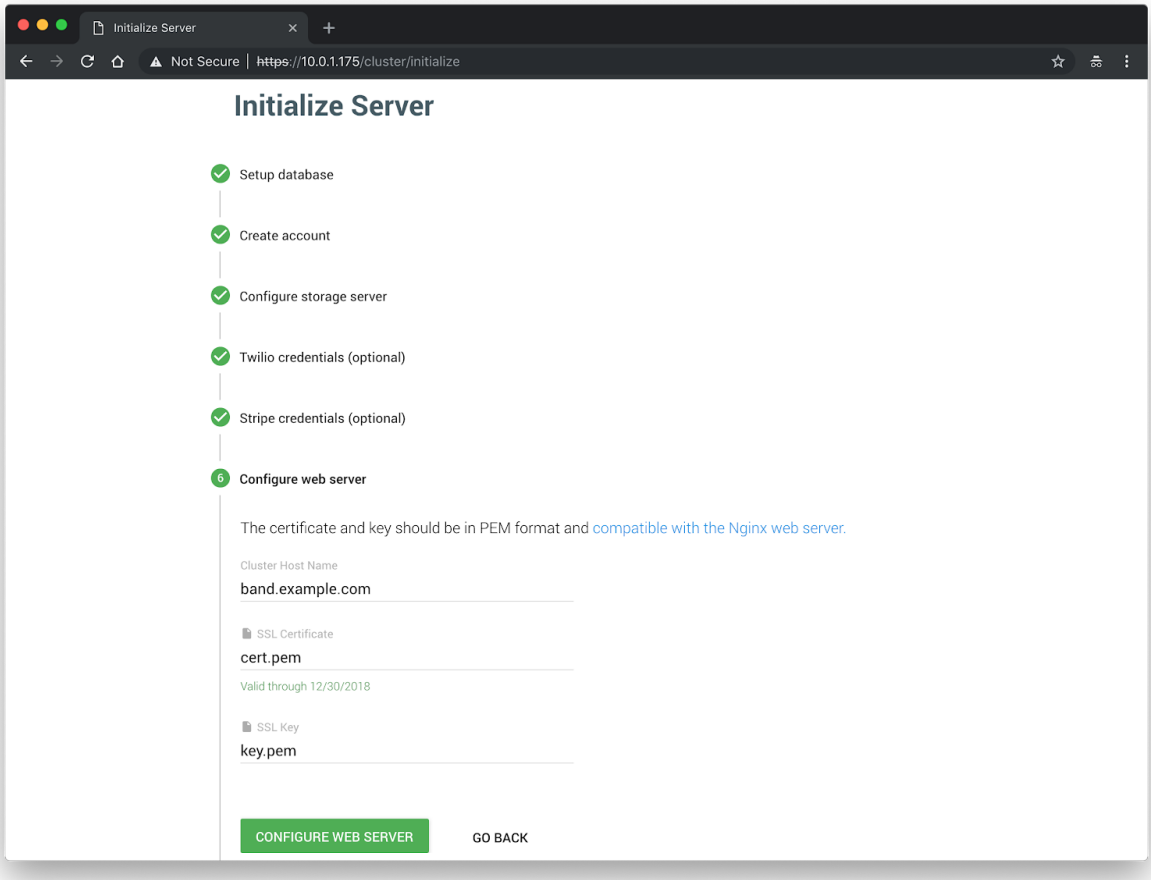
From **Initialize Server: Twilio credentials**, configure a Twilio account with Band in order to allow users to log in using their mobile numbers by default. However, team administrator can disable this feature on Bridge by configuring alternative authentication mechanism.



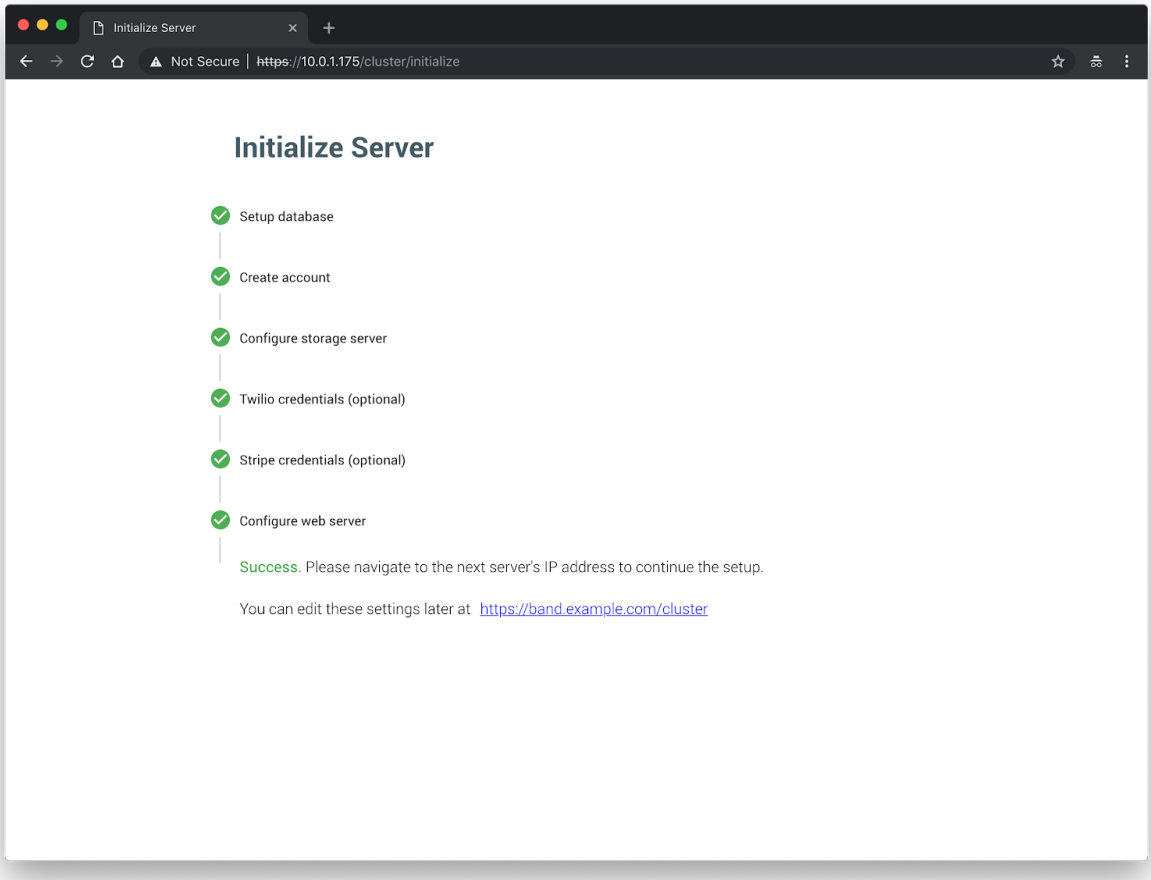
From **Initialize Server: Stripe credentials**, this step is not required, click on **Skip** button to skip this step.



From **Initialize Server: Configure web server**, enter the cluster hostname and associated SSL certificate and keys. These files should be compatible with the Nginx web server.

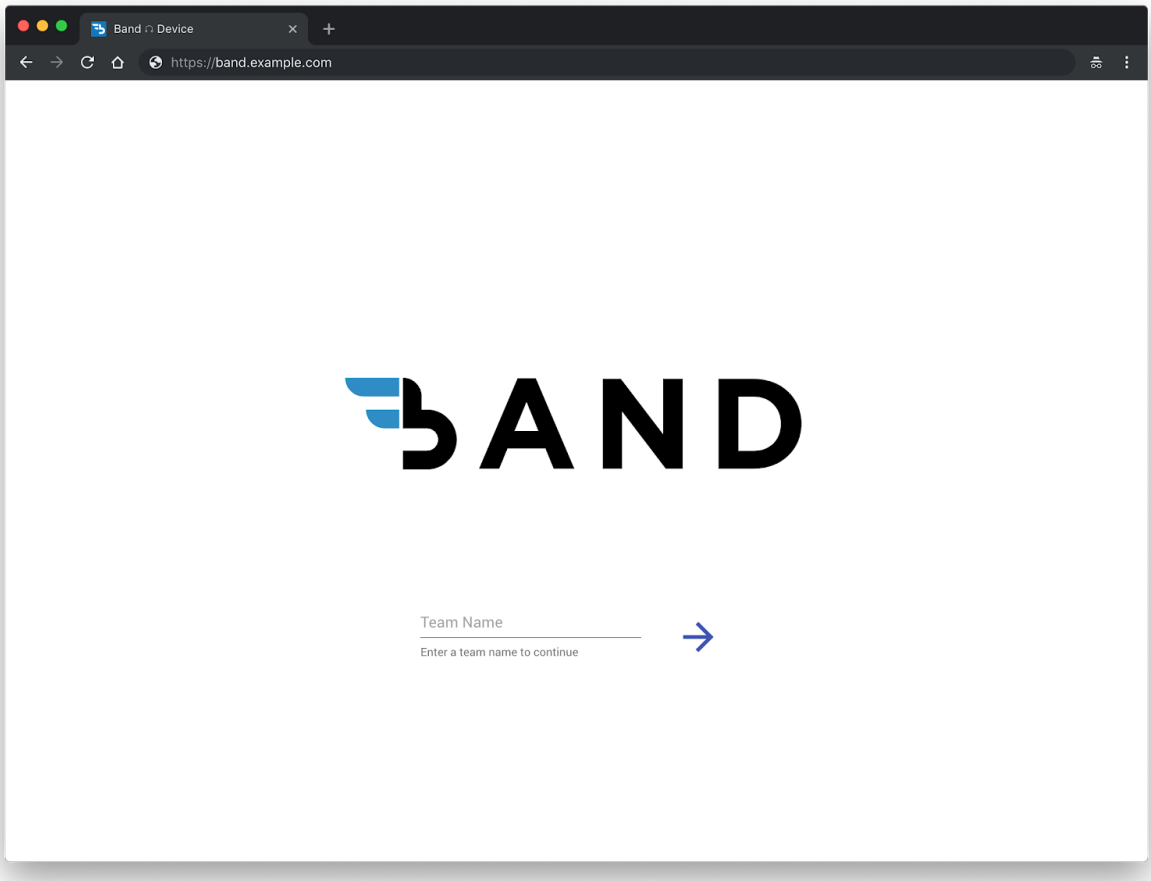


After completion, navigate to the next server's IP address to continue the setup. You can also click the link to navigate to server settings.



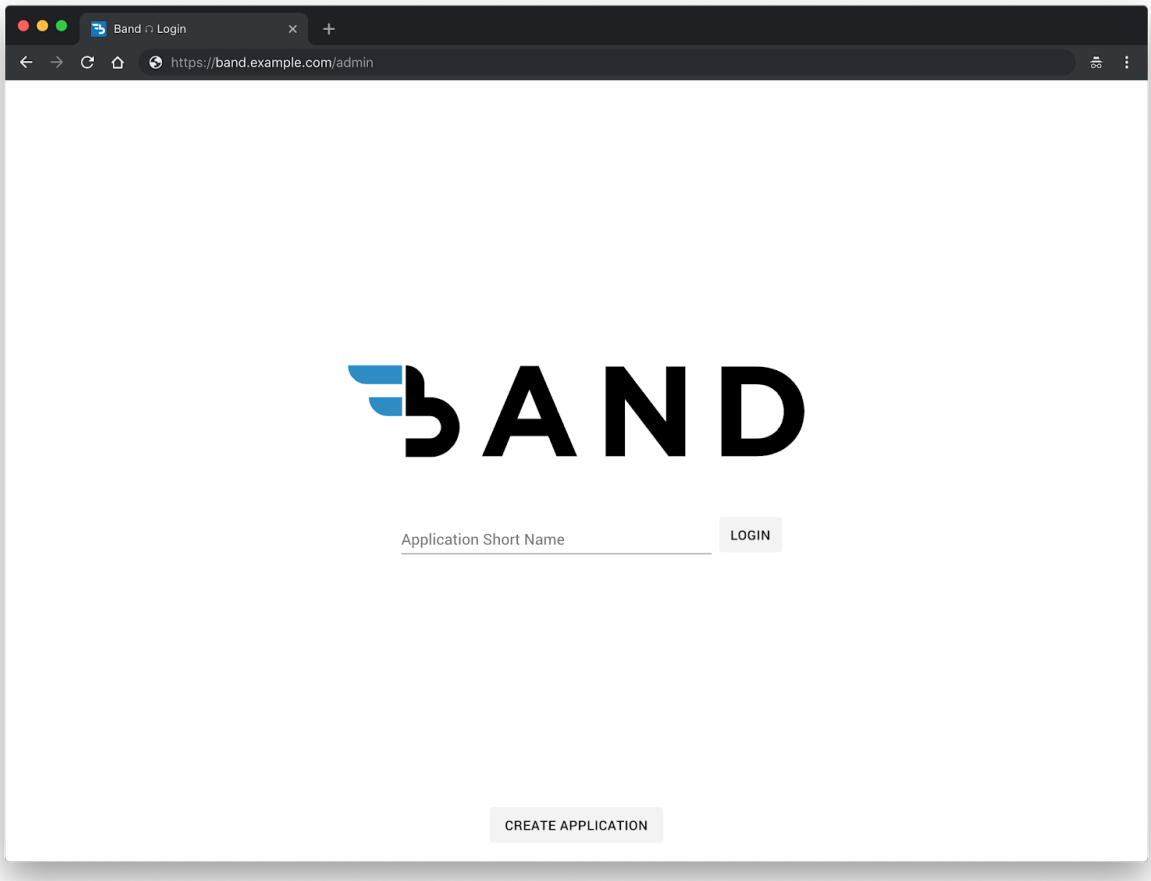
Verify DNS and SSL

Navigate to the HTTPS designated cluster hostname to verify setup. If the cluster hostname were **band.example.com,** the address would be **https://band.example.com**.



Client Administration

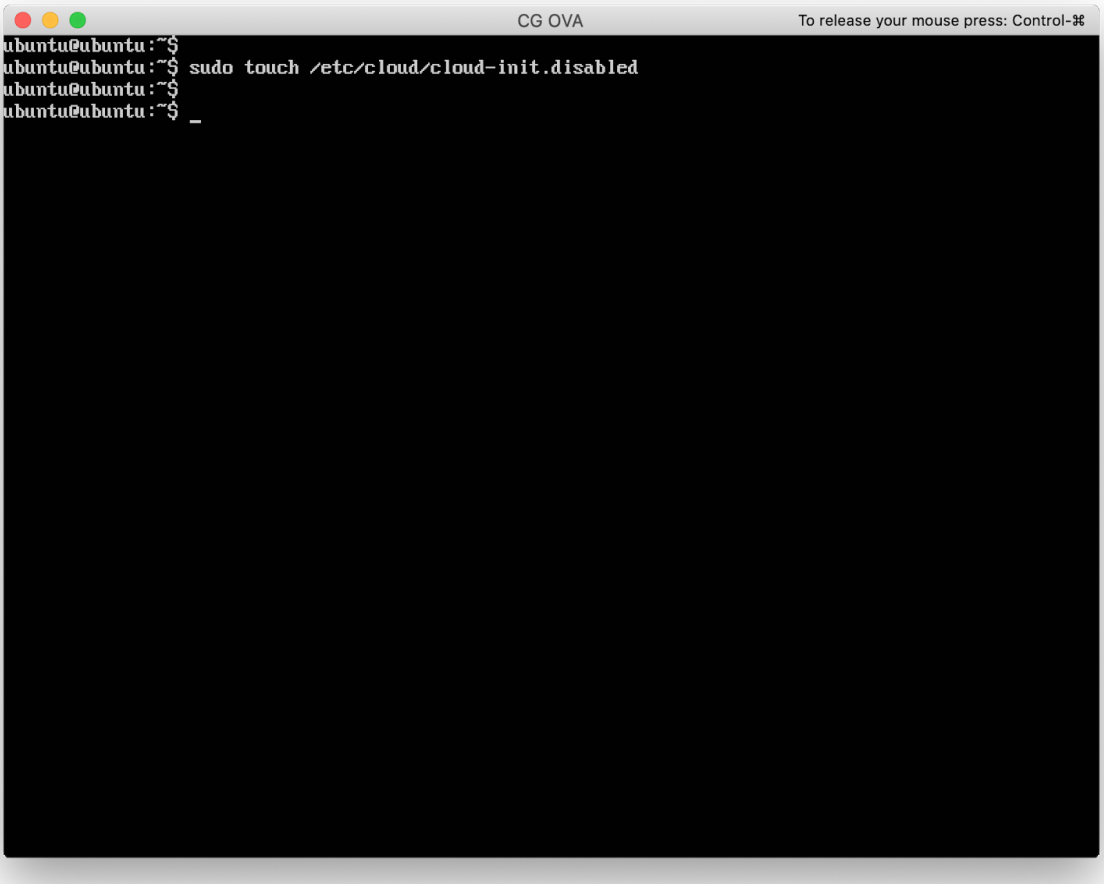
To access the client site, navigate to the **/admin** route. If the cluster hostname were **band.example.com,** the address would be **https://band.example.com/admin.**



Disable Cloud-Init

Cloud-init is the service that initializes cloud images on EC2. However, it is not required when running the server on-premise. Disable cloud-init by running the following command,

```
sudo touch /etc/cloud/cloud-init.disabled
```

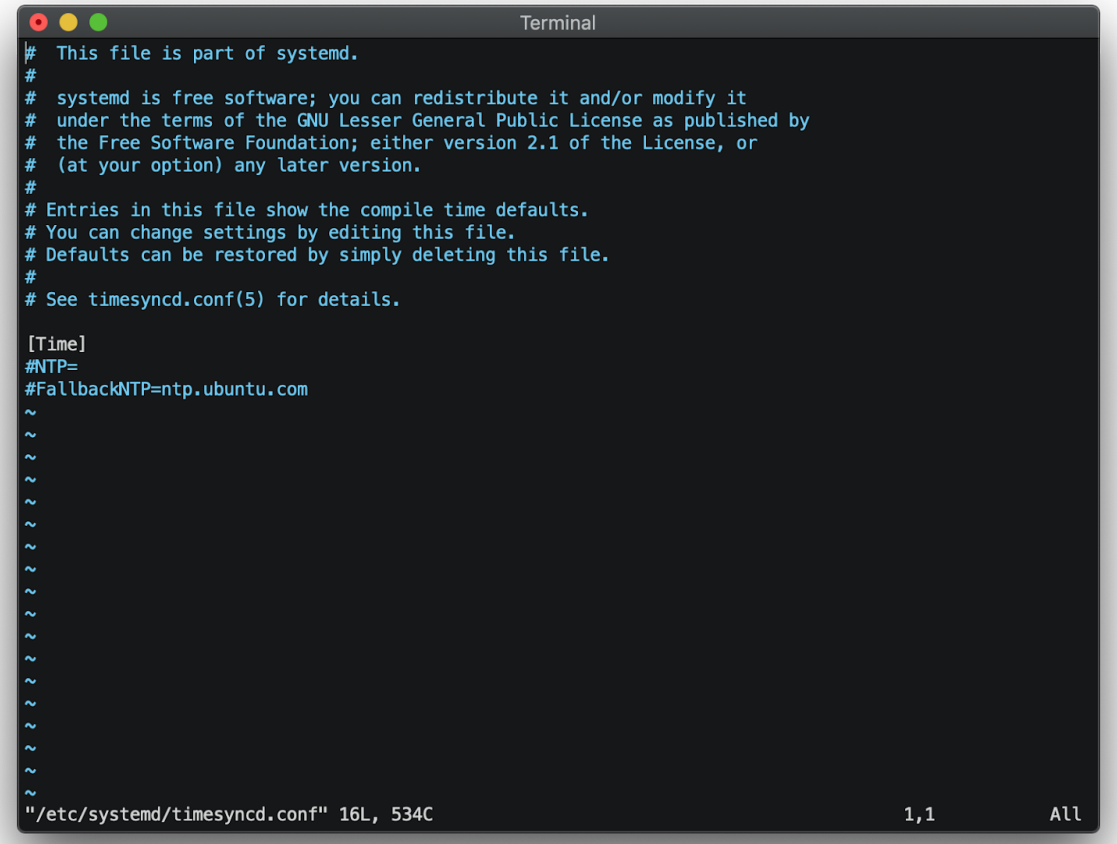


Set up Custom Timeserver (Optional)

Ubuntu 16.04 by default uses the time server at **ntp.ubuntu.com.**

To change the default server, paste the following command in the terminal to edit the configuration file in the VIM editor:

```
sudo vim /etc/systemd/timesyncd.conf
```



Uncomment the NTP line by removing the hash and enter the desired time server address. For example, if the NTP time server was **time.google.com**, the entry would be **NTP=time.google.com**

Quit the editor by hitting **ESC**, and type **:wq** to save and exit. Hit **Enter**. Restart the time server by running the following command in the terminal:

```
systemctl restart systemd-timesyncd
```

To check the status of the timeserver, run the following command in the terminal:

```
systemctl status systemd-timesyncd
```