

Configuration



- [Gateway Configuration Block](#)
 - [Gateway Entry](#)
 - [TLS Entry - Certificate rotation](#)
 - [Gateway behind Load Balancer](#)
-

The `gateway` [configuration block](#) is similar to a `cluster` block:

```
gateway {
  name: "A"
  listen: "localhost:7222"
  authorization {
    user: gwu
    password: gwp
  }

  gateways: [
    {name: "A", url: "nats://gwu:gwp@localhost:7222"},
    {name: "B", url: "nats://gwu:gwp@localhost:7333"},
    {name: "C", url: "nats://gwu:gwp@localhost:7444"},
  ]
}
```

One difference is that instead of `routes` you specify `gateways`. As expected *self-gateway* connections are ignored, so you can share gateway configurations with minimal fuss.

Starting a server:

```
nats-server -c A.conf
```

```
[85803] 2019/05/07 10:50:55.902474 [INF] Starting nats-server version 2.0.0
[85803] 2019/05/07 10:50:55.903669 [INF] Gateway name is A
[85803] 2019/05/07 10:50:55.903684 [INF] Listening for gateways connections on localhost:7222
```

```
[85803] 2019/05/07 10:50:55.903696 [INF] Address for gateway "A" is localhost:7222
[85803] 2019/05/07 10:50:55.903909 [INF] Listening for client connections on 0.0.0.0:4222
[85803] 2019/05/07 10:50:55.903914 [INF] Server id is
NBHUDBF3TVJSWCDPG2HSKI4I2SBSPDTNYEXEM0FAZUZYXVA2IYRUGPZU
[85803] 2019/05/07 10:50:55.903917 [INF] Server is ready
[85803] 2019/05/07 10:50:56.830669 [INF] 127.0.0.1:50892 - gid:2 - Processing inbound gateway
connection
[85803] 2019/05/07 10:50:56.830673 [INF] 127.0.0.1:50891 - gid:1 - Processing inbound gateway
connection
[85803] 2019/05/07 10:50:56.831079 [INF] 127.0.0.1:50892 - gid:2 - Inbound gateway connection
from "C" (NBHWDF03KHANNI6UCEUL27VNW7NWD2MC4BI4L2C7VVLFBMZ3CRD7HE) registered
[85803] 2019/05/07 10:50:56.831211 [INF] 127.0.0.1:50891 - gid:1 - Inbound gateway connection
from "B" (ND2UJB3GFUHX0Q2UUMZQG0CL4QVR2LRJ0DPZH7MIPGLWCQRARJBU27C3) registered
[85803] 2019/05/07 10:50:56.906103 [INF] Connecting to explicit gateway "B" (localhost:7333)
at 127.0.0.1:7333
[85803] 2019/05/07 10:50:56.906104 [INF] Connecting to explicit gateway "C" (localhost:7444)
at 127.0.0.1:7444
[85803] 2019/05/07 10:50:56.906404 [INF] 127.0.0.1:7333 - gid:3 - Creating outbound gateway
connection to "B"
[85803] 2019/05/07 10:50:56.906444 [INF] 127.0.0.1:7444 - gid:4 - Creating outbound gateway
connection to "C"
[85803] 2019/05/07 10:50:56.906647 [INF] 127.0.0.1:7444 - gid:4 - Outbound gateway connection
to "C" (NBHWDF03KHANNI6UCEUL27VNW7NWD2MC4BI4L2C7VVLFBMZ3CRD7HE) registered
[85803] 2019/05/07 10:50:56.906772 [INF] 127.0.0.1:7333 - gid:3 - Outbound gateway connection
to "B" (ND2UJB3GFUHX0Q2UUMZQG0CL4QVR2LRJ0DPZH7MIPGLWCQRARJBU27C3) registered
```

Once all the gateways are up, these clusters of one will forward messages as expected:

```
nats sub -s localhost:4333 ">"
```

On a different session...

```
nats pub -s localhost:4444 foo bar
```

The subscriber should print

```
[#1] Received on [foo] : 'bar'
```

Gateway Configuration Block

Property	Description	Version
<code>name</code>	Name for this cluster, all gateways belonging to the same cluster, should specify the same name.	
<code>reject_unknown_cluster</code>	If <code>true</code> , gateway will reject connections from cluster that are not configured in <code>gateways</code> . It does so by checking if the cluster name, provided by the incoming connection, exists as named gateway. This effectively disables gossiping of new cluster. It does not restrict a configured gateway, thus cluster, from dynamically growing.	
<code>gateways</code>	List of Gateway entries - see below.	
<code>host</code>	Interface where the gateway will listen for incoming gateway connections.	
<code>port</code>	Port where the gateway will listen for incoming gateway connections.	
<code>listen</code>	Combines <code>host</code> and <code>port</code> as <code><host>:<port></code>	
<code>tls</code>	A tls configuration map for securing gateway connections. <code>verify</code> is always enabled. Unless otherwise specified in a gateway , <code>cert_file</code> will be the default client certificate. Certificate pitfalls .	
<code>advertise</code>	Hostport <code><host>:<port></code> to advertise how this server can be contacted by other gateway members. This is useful in setups with NAT, or in the cloud when exposed by a Network Load Balancer.	
<code>connect_retries</code>	After how many failed connect attempts to give up establishing a connection to a discovered gateway. Default is <code>0</code> , do not retry. When enabled, attempts will be made once a second. This, does not apply to explicitly configured gateways.	
<code>connect_backoff</code>	Enable exponential backoff on reconnection attempts. If <code>true</code> , will start exponential backoff at 1 second up to 30 seconds.	2.12.0

Property	Description	Version
<code>authorization</code>	Authorization map for gateways. When a single <code>username / password</code> is used, it defines the authentication mechanism this server expects, and how this server will authenticate itself when establishing a connection to a <i>discovered</i> gateway. This will not be used for gateways explicitly listed in gateways and therefore have to be provided as part of the URL. With this authentication mode, either use the same credentials throughout the system or list every gateway explicitly on every server. If the <code>tls</code> configuration map specifies <code>verify_and_map</code> only provide the expected <code>username</code>. Here different certificates can be used, but they do have to map to the same <code>username</code>. The authorization map also allows for <code>timeout</code> which is honored but <code>users</code> and <code>token</code> configuration are not supported and will prevent the server from starting. The <code>permissions</code> block is ignored.	

Gateway Entry

The `gateways` configuration block is a list of gateway entries with the following properties:

Property	Description
<code>name</code>	Gateway name.
<code>url</code>	Hostport <code><host>:<port></code> describing where the remote gateway can be reached. If multiple IPs are returned, one is randomly selected.
<code>urls</code>	A list of <code>url</code> strings.
<code>tls</code>	A tls configuration map for creating a secure gateway connection. If the top-level <code>gateway{}</code> <code>tls</code> block contains certificates that have both client and server purposes, it is possible to omit this one and the server will use the certificates from the <code>gateway{tls{}}</code> section. See additional advice below in TLS Entry.

By using `urls` and an array, you can specify a list of endpoints that form part of a cluster as below. A NATS Server will pick one of those addresses randomly and only establish a single outbound gateway connection to one of the members from another cluster:

```
gateway {
  name: "DC-A"
  listen: "localhost:7222"

  gateways: [
    {name: "DC-A", urls: ["nats://localhost:7222", "nats://localhost:7223",
"nats://localhost:7224"]},
    {name: "DC-B", urls: ["nats://localhost:7332", "nats://localhost:7333",
"nats://localhost:7334"]},
    {name: "DC-C", urls: ["nats://localhost:7442", "nats://localhost:7333",
"nats://localhost:7335"]}
  ]
}
```

TLS Entry - Certificate rotation

In addition to the normal TLS configuration advice, bear in mind that TLS keys and certificates for multiple clusters, or servers in different locations, rarely rotate at the exact same time and that Certificate Authorities do roll between multiple Intermediate certificates. Make sure that the `ca_file` contains both old and new root or intermediate CAs.

If using a certificate bundle which accompanied the issuance of a certificate then the CA in that bundle will typically be for just that certificate. Using *only* that CA as the CA for gateway authentication is ill-advised. You should ensure that you allow for rolling between Certificate Authorities, even if only between multiple CAs from the same organization entity, and use a separate certificate bundle for *verification* of peers. This way when DC-B rolls before DC-A, it will not be cut off from your supercluster.

To apply changes (even when only changing the content of those files) the server configuration needs to be reloaded. [Configuration reloading](#)

Gateway behind Load Balancer

When running in a private network (such as VPC in the cloud), you'd probably want to setup a Load Balancer to expose the gateway port. In this case, it's important to set the `advertise` value of each NATS server to the Load Balancer hostname and port serving them. Otherwise, they will advertise their own private IP and port, and can generate failures and downtime if any reconnection is required.

📄 #3

📄 cocobook-sync 📄 2026-10-28 11:13:50 UTC

📄 cocobook-sync 📄 2026-10-29 01:42:38 UTC