

# Polkadot Gateway

- [Overview](#)
- [Architecture](#)
- [Manage using marlinctl](#)
- [Manage manually](#)

# Overview

The polkadot gateway acts as a bridge between your polkadot node and the Marlin relay network, enabling them to understand and talk to each other. In addition, it handles most of the Marlin related interactions for you like optimizing peer sets, attesting messages, generating tickets, etc.

## Quickstart

This section assumes that marlinctl is installed already. See [here](#) if you have not installed it.

**Step 1:** Generate a client identity for the gateway by running the below command and entering a passphrase when prompted

```
sudo marlinctl gateway dot keystore create
```

**Step 2:** Create the gateway (currently using staking data from kovan)

```
sudo marlinctl gateway dot create --bootstrap-addr "54.193.224.20:8002"
```

**Step 3:** Retrieve the polkadot identity of the gateway by checking the logs

```
sudo marlinctl gateway dot logs
```

Look for a line like `Local node identity is: <IDENTITY>`, the alphanumeric string at the end is the identity.

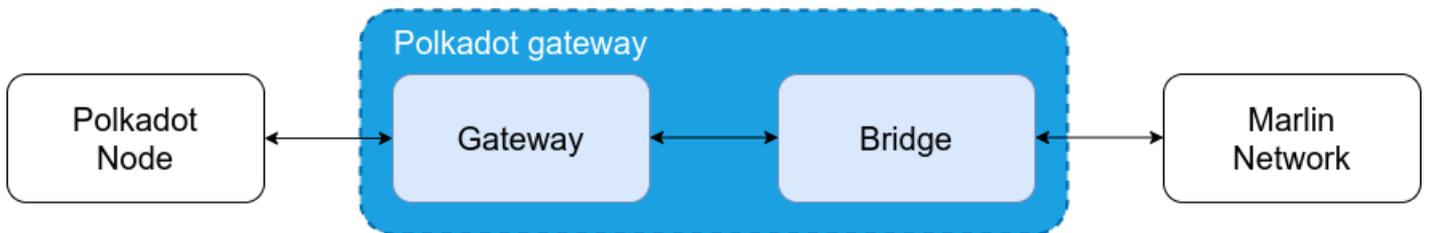
**Step 4:** Add the gateway as a persistent peer to your polkadot node by passing it as a `--reserved-nodes` parameter

```
# Use the identity obtained above
polkadot --reserved-nodes "/ip4/<GATEWAY_IP>/tcp/20900/p2p/<IDENTITY>"
```

That's it!



# Architecture



The polkadot gateway is comprised of two components

- **Gateway** - Responsible for managing connections on the polkadot side and translating between the polkadot wire protocol and an intermediate format.
- **Bridge** - Responsible for managing connections on the Marlin side and translating between the intermediate format and the Marlin wire protocol. Also responsible for managing Marlin specific bits like discovery and message attestations.

# Manage using marlinctl

This page assumes that `marlinctl` is installed already. See [here](#) if you have not installed it.

## Command tree

The `marlinctl` command subtree for managing Polkadot gateways resides under `marlinctl gateway dot`. As with all `marlinctl` commands, the subtree can be incrementally discovered with the `--help` parameter.

```
$ sudo marlinctl gateway dot --help
```

Polkadot Gateway

Usage:

```
marlinctl gateway dot [command]
```

Available Commands:

<code>config</code>	Configurations of project set on disk
<code>create</code>	Create gateway for polkadot blockchain
<code>destroy</code>	Destroy gateway for polkadot blockchain
<code>logs</code>	Tail logs for running gateway (polkadot) instances
<code>recreate</code>	Recreate end to end gateway (polkadot) instances
<code>restart</code>	Restart services for gateway (polkadot) instances
<code>status</code>	Show status of currently running gateway (polkadot) instances
<code>versions</code>	Show available versions for use

Flags:

```
-h, --help help for dot
```

Global Flags:

```
--config string          config file (default is $HOME/.marlin/ctl/state.yaml)
--forceful-registry-sync forceful registry sync. Do not use if you don't know what
this is for.
```

```
--loglevel string      marlinctl loglevel (default is INFO) (default "info")
--skip-registry-sync  skip registry sync during run
--skip-update-check   skip update check during run
```

Use "marlinctl gateway dot [command] --help" for more information about a command.

## Create a gateway

Gateways can be created using the `create` command.

```
$ sudo marlinctl gateway dot create --help
```

Create gateway for polkadot blockchain

Usage:

```
marlinctl gateway dot create [flags]
```

Flags:

```
-b, --bootstrap-addr string      Bridge bootstrap address
-a, --chain-identity string      Gateway's keystore path (default "gateway_dot.key")
-c, --contracts string          mainnet/kovan (default "mainnet")
-d, --discovery-addr string      Bridge discovery address (default "0.0.0.0:20702")
-h, --help                      help for create
-i, --instance-id string         instance-id of spawned up resource (default "001")
-l, --internal-listen-address string Bridge listen address (default "127.0.0.1:20901")
-y, --keystore-pass-path string  Keystore pass path
-k, --keystore-path string      Keystore Path
-g, --listen-addr string         Address on which gateway listens for connections
from peer (default "/ip4/0.0.0.0/tcp/20900")
-p, --pubsub-addr string         Bridge pubsub address (default "0.0.0.0:20700")
-r, --runtime-args stringToString runtime arguments while starting up (default [])
-s, --skip-checksum             skip checksum verification while starting up
binaries
-x, --version string            runtime version override
```

Global Flags:

```
--config string      config file (default is $HOME/.marlin/ctl/state.yaml)
```

```
--forceful-registry-sync    forceful registry sync. Do not use if you don't know what this is
for.
  --loglevel string          marlinctl loglevel (default is INFO) (default "info")
  --skip-registry-sync      skip registry sync during run
  --skip-update-check       skip update check during run
```

## Primary flags

While the following flags are optional for testing the connection between the gateway and the polkadot node, they are required parameters for interacting with the Marlin network.

- `--bootstrap-addr` - Address of the bootstrap server used for discovering other nodes in the network.
- `--keystore-path` - Path to keystore file. Serves as the identity of the gateway.
- `--keystore-pass-path` - Path to text file containing passphrase that can unlock the above keystore file.

## Secondary flags

- `--contracts` - Contract set used by the gateway for fetching on-chain data. Enables choosing the network that the gateway is a part of (e.g. separate networks for testing or new upgrades). Especially important in the initial phases of the network, where it's expected that clusters will work on a testnet first before transitioning to mainnet upon maturity.
- `--chain-identity` - Path to a polkadot key file which the gateway uses as its identity. Useful to give the gateway a deterministic identity on the polkadot side so deployments can be automated.

The following flags are useful for customizing the socket addresses used by the gateway in order to prevent any conflicts with other programs on the same system.

- `--listen-addr` - Address where the gateway listens for incoming connections from your polkadot nodes.
- `--discovery-addr` - Address used for the discovery protocol to find peers in the Marlin network.
- `--pubsub-addr` - Address used for the pubsub protocol, to send and receive message from the Marlin network.
- `--internal-listen-addr` - Address used for internal communication between components of

the gateway.

## See logs

Logs can be tailed using the `logs` command.

```
$ sudo marlinctl gateway dot logs --help
```

Tail logs for running gateway (polkadot) instances

Usage:

```
marlinctl gateway dot logs [flags]
```

Flags:

```
-h, --help                help for logs
-i, --instance-id string  instance-id of resource to log (default "001")
-n, --last int            number of last lines to tail in logfile (default 100)
```

Global Flags:

```
--config string          config file (default is $HOME/.marlin/ctl/state.yaml)
--forceful-registry-sync forceful registry sync. Do not use if you don't know what
this is for.
--loglevel string        marlinctl loglevel (default is INFO) (default "info")
--skip-registry-sync     skip registry sync during run
--skip-update-check      skip update check during run
```

## Primary flags

- `--last` - Number of lines from the end to tail

## Destroy a gateway

Gateways can be destroyed using the `destroy` command.

```
$ sudo marlinctl gateway dot destroy --help
```

Destroy gateway for polkadot blockchain

Usage:

```
marlinctl gateway dot destroy [flags]
```

Flags:

```
-h, --help                help for destroy  
-i, --instance-id string  instance-id of resource to destroy (default "001")
```

Global Flags:

```
--config string           config file (default is $HOME/.marlin/ctl/state.yaml)  
--forceful-registry-sync  forceful registry sync. Do not use if you don't know what  
this is for.  
--loglevel string         marlinctl loglevel (default is INFO) (default "info")  
--skip-registry-sync     skip registry sync during run  
--skip-update-check      skip update check during run
```

# Manage manually

## Prebuilt binaries

Prebuilt binaries for the components are available here:

gateway	
bridge	

## Running the bridge

The bridge parameters can be discovered with the `--help` flag.

```
$ ./bridge_dot_linux-amd64 --help

USAGE: bridge [OPTIONS]

OPTIONS:
  -d, --discovery-addr <discovery_addr>
  -p, --pubsub-addr <pubsub_addr>
  -b, --beacon-addr <beacon_addr>
  -l, --listen-addr <listen_addr>
  -k, --keystore-path <keystore_path>
  -k, --keystore-pass_path <keystore_pass_path>
  -c, --contracts <contracts>
  -h, --help <help>
  -v, --version <version>
```

## Primary flags

While the following flags are optional for testing the connection between the gateway and the bridge, they are required parameters for interacting with the Marlin network.

- `--beacon-addr` - Address of the bootstrap server used for discovering other nodes in the

network.

- `--keystore-path` - Path to keystore file. Serves as the identity of the bridge.
- `--keystore-pass-path` - Path to text file containing passphrase that can unlock the above keystore file.

## Secondary flags

- `--contracts` - Contract set used by the bridge for fetching on-chain data. Enables choosing the network that the bridge is a part of (e.g. separate networks for testing or new upgrades). Especially important in the initial phases of the network, where it's expected that clusters will work on a testnet first before transitioning to mainnet upon maturity.

The following flags are useful for customizing the socket addresses used by the bridge in order to prevent any conflicts with other programs on the same system.

- `--discovery-addr` - Address used for the discovery protocol to find peers in the Marlin network.
- `--pubsub-addr` - Address used for the pubsub protocol, to send and receive message from the Marlin network.
- `--listen-addr` - Address used for listening to connections from the gateway component.

## Running the gateway

The gateway parameters can be discovered with the `--help` parameters.

```
$ ./gateway_dot_linux-amd64 --help
gateway_dot 0.1.0

USAGE:
  gateway_dot_linux-amd64 [OPTIONS]

FLAGS:
  -h, --help          Prints help information
  -V, --version       Prints version information

OPTIONS:
```

```
-b, --bridge-addr <bridge-addr>
-k, --keystore-path <keystore-path>
-l, --listen-addr <listen-addr>
```

## Primary flags

- `--bridge-addr` - Address of the bridge.

## Secondary flags

- `--listen-addr` - Address used for listening to incoming connections from your polkadot nodes.
- `--keystore-path` - Path to a polkadot key file which the gateway uses as its identity. Useful to give the gateway a deterministic identity on the polkadot side so deployments can be automated.