

Deploying to the Live Network

When you're finished developing your contracts and would like others to use them, the next step is to deploy them to the live network. By now you will have been working on a development network like [Ganache](#) , and you will have noticed Truffle is set up to support that by default. Deploying to the live network -- or any other network -- will require that you first connect one of the [many Ethereum clients](#) to your network of choice, as well as configure Truffle correctly.

Setting Up Your Ethereum client¶

There are [many Ethereum clients](#) to choose from. The most popular are [go-ethereum](#) and [cpp-ethereum](#) . This tutorial cannot go into the details of setting up each client for each operating system, but be sure your desired client is installed and configured to perform the following:

- Your client is completely synced with the live network
- Your client is hosting an RPC server on 127.0.0.1
- and port 8546
- (for this tutorial)
- Your client has at least one account registered and can sign transactions for that account
- The registered account contains enough Ether to deploy your contracts

If you don't want to set up an Ethereum client yourself, you can alternatively use [Infura](#) . This will require you [configure Truffle correctly for Infura](#) before proceeding.

Configuring Truffle¶

The default Truffle configuration without any bells and whistles looks like this:

```
module . exports
```

```
=
```

```
{
```

```
  rpc :
```

```
  {
```

```
    host :
```

```
    "127.0.0.1" ,
```

```
    port :
```

```
    8545
```

```
  } };
```

This tells Truffle that by default it should connect to an Ethereum client at host 127.0.0.1 and port 8545 . You could keep this configuration and simply stop your development client and run the live network at the same host and port; however, Truffle won't know how to distinguish deployment artifacts on the development network from those same artifacts on the live network (like deployed addresses, for instance). To ensure Truffle knows the network you want to deploy to, we can add a specific configuration for the live network:

```
module . exports
```

```
=
```

```
{
```

```
  networks :
```

```
  {
```

```
    "live" :
```

```
    {
```

```
      network_id :
```

```
      1 ,
```

```
      host :
```

```
"127.0.0.1" ,
```

```
port :
```

```
8546
```

```
// Different than the default below
```

```
}
```

```
},
```

```
rpc :
```

```
{
```

```
host :
```

```
"127.0.0.1" ,
```

```
port :
```

```
8545
```

```
}}; Notice we set up our live Ethereum client to run on a different port to ensure we don't inadvertently deploy to the wrong network.
```

One important configuration option to pay attention to is `network_id` . Here, we set the network id to 1 to signify that the network named "live" within the configuration represents only the network identified by an id of 1 . Since only the live Ethereum network has that id (all other networks do not), we can be sure this configuration will only apply to the live Ethereum network. Private networks will have their own network id, for example, and the [Morden testnet](#) -- a publically shared network used only for testing -- has a network id of 2 .

Deploying to the Live Network ¶

Now that our configuration is set up, we can now deploy to the live network. When deploying, we must specifically ask for the live network or else Truffle will fall back to the default configuration. To do so, we can deploy with the following command:

```
truffle migrate --network live
```

 Notice that we asked for the "live" network, which is the name we defined in the configuration, and Truffle will connect to the specified host and port -- in this case `http://127.0.0.1:8546` -- to deploy our contracts.

Your [migrations](#) are run on this network just as they'd run on any other network. If you had previously deployed to the live network before, `truffle migrate --network live` will check which migration was last run and only start deploying from there.

And That's it! □

Congrats! You have now deployed your contracts to the live network. For more details on network configuration and deployment, be sure to check out the [Truffle documentation](#) .