Elastos × Bitcoin Merged Mining: Step-by-Step

1 — Prepare the Elastos node

1. **Get the files**

```
None
wget https://download.elastos.io/elastos-ela/elastos-ela-v0.9.9
wget
https://download.elastos.io/elastos-ela/elastos-ela-v0.9.9/sponso
rs
```

2. **Create config.json** in the same folder:

```
None
{
  "Configuration": {
    "PrintLevel": 0,
    "PowConfiguration": {
      "PayToAddr": "8VYXVxKKSAxkmRrfmGpQR2Kc66XhG6m3ta",
      "MinerInfo": "ELA",
      "MinTxFee": 100
    },
    "EnableRPC": true,
    "RpcConfiguration": {
      "User": "yourRpcUser",
      "Pass": "yourRpcPass",
      "WhiteIPList": ["0.0.0.0"]
    },
    "DPoSConfiguration": {
      "SponsorsFilePath": "./sponsors"
    }
  }
}
```

- PayToAddr — your ELA address for block rewards
- User/Pass + WhiteIPList — basic RPC access control
- SponsorsFilePath — points to the downloaded **sponsors** file

3. **Place the sponsors file** in the folder you will run the node from (or keep the path above).
4. **Start the node**

```
None
./ela
```

The first launch sets up data folders, then the chain sync begins. Let it reach the tip before mining.

---

2 — Pool-side calls

Once the node is synced and reachable on its RPC port:

Call	Purpose
createauxblock	Ask Elastos for a new work template
submitauxblock	Send the solved **auxpow** back

2.1 Create work

Request (named args):

```
None
{
  "method": "createauxblock",
  "params": { "paytoaddress":
"Ef4UcaHwvFrFzzsyVf5YH4JBWgYgUqfTAB" }
}
```

Response (trimmed):

```
None
{
  "result": {
    "hash": "e28a262b3831…",
    "height": 152789,
    "coinbasevalue": 175799086,
    "bits": "1d36c855",
    "chainid": 1224,
    "previousblockhash": "f297d03791f4…"
  }
}
```

Keep the hash; you'll need it in step 2.2.

2.2 Submit the solved block

Request (named args):

```
None
{
  "method": "submitauxblock",
  "params": {
    "blockhash": "e28a262b3831…",
    "auxpow": "0200000001…a880495b"
  }
}
```

Successful reply:

```
None
{ "result": true }
```

---

3 — Full mining loop

1. Pool calls **createauxblock**.
2. Pool builds a Bitcoin block header that embeds the hash from step 1.
3. Bitcoin miners search for a valid Bitcoin proof-of-work.
4. When found, pool assembles the **auxpow** string.
5. Pool calls **submitauxblock** with the original hash and the new **auxpow**.
6. Elastos validates the proof and pays the reward to PayToAddr.

Repeat the loop for each new Bitcoin block template.