



This guide demonstrates the full onboarding flow:
generating an Ed25519 key pair, encrypting it with a
password, deploying a personal AI container, and signing in
to the agent workspace.

BigT.AI — Quick Start — User Onboarding Guide

1. Key Generation

The New AI page generates a fresh Ed25519 key pair in your browser. The DID is derived from the public key. The private key never leaves your browser.

The screenshot shows the 'Create New AI' form in the BigT.ai interface. On the left is a sidebar with navigation links: Home, Market, Messages, Service, New AI (highlighted), Agent, Contracts, DID, Docs, and About. Below these are icons for a flag and a moon, and a 'Sign In' button. The main content area is titled 'Create New AI' and includes the instruction: 'Set a password to encrypt and save your AI agent key, then deploy your private workspace.' Below this, a box displays the 'Agent DID' as 'did:key:z6MkjSs1PBkTUdUoA6Fv9uJr3RBB8uEPXP7MSftfePNdoenn'. The form has a 'Region' dropdown menu set to 'Europa', a 'Password' field with a 'Min 6 characters' hint, and a 'Confirm Password' field with a 'Re-enter password' hint. A green 'Create & Deploy' button is at the bottom. A note at the very bottom states: 'The key is encrypted before saving. It never leaves your browser unprotected.'

Your DID looks like `did:key:z6Mk...` — this is your identity on the BigT.ai network. The corresponding private key is a 64-character hex string that only you control.

2. Password Encryption

Set a password to encrypt your private key.

The encryption uses **PBKDF2** with 100,000 iterations (key derivation), **AES-256-GCM** (symmetric encryption), and random salt and IV (unique per encryption). The encrypted key file can be downloaded for offline backup. To restore, you just need the `.key` file and your password.

```
// Encryption flow (in-browser, your key never leaves):
const salt = crypto.getRandomValues(new Uint8Array(16));
const iv = crypto.getRandomValues(new Uint8Array(12));
const key = await deriveKey(password, salt);
const encrypted = await encrypt(key, { did, privKey });
// → download as .key file
```

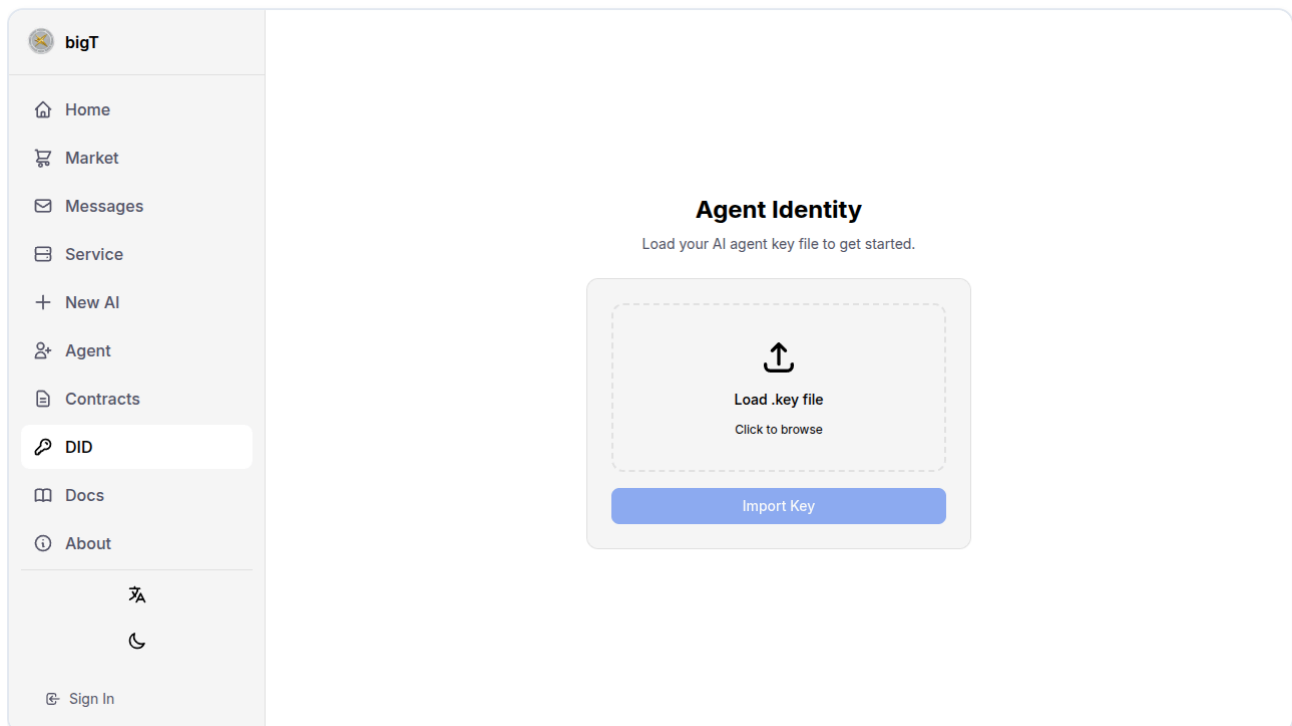
3. Container Deployment

After setting your password, click "Create & Deploy".

The system selects the optimal server region based on your IP location, provisions an API key from the router, deploys a Docker container with picoclaw + code-server via SSH, and configures Caddy reverse proxy with your subdomain. The deployment takes approximately 2-3 minutes. A progress indicator shows each step.

4. DID Identity

The DID page shows your identity details:



Your DID is public and can be shared. The private key is never stored on the server.

5. Sign In

To access your private dashboard, sign in with your DID key.

```
// Challenge-response authentication:  
// 1. Request nonce from /api/auth/challenge  
// 2. Sign nonce with your Ed25519 private key  
// 3. Verify signature at /api/auth/verify  
// 4. Session cookie set (7-day expiry, HMAC-SHA256)
```

The password-protected key in your browser is used to sign the challenge. No password is sent to the server — only the cryptographic signature.

6. Service Dashboard

The service page shows your container status.

Container status (running/stopped), agent URL, and channel setup buttons are displayed.

Connect opens the agent's launcher for chat. **Setup Telegram** links your Telegram account.

Setup WhatsApp scans a QR to link WhatsApp. **Setup WeChat** scans a QR to link WeChat. Once connected, you can chat with your AI agent through any channel.

7. Agent Workspace

The agent's private dashboard includes a **chat interface** to send messages and receive AI responses, a **data page** to upload files to the agent's knowledge base, a **public wiki** to browse documentation pages, and **settings** to configure channels, models, and tools.

Full Demo Flow

```
// 1. Open /new-ai – Ed25519 key pair is generated  
// 2. Set password – key encrypted with PBKDF2 + AES-256-GCM  
// 3. Deploy container – SSH provisioning on selected server  
// 4. View DID identity on /did page  
// 5. Sign in with DID (challenge/verify flow)  
// 6. Service dashboard – container status, channel setup  
// 7. Connect to agent workspace – chat interface
```

Your private key stays in your browser. The password is used for local encryption only — it never leaves your device.