

# Candy Machine v3 Deployment Guide — Custom Mini Rooms

This guide walks your team step-by-step through preparing and deploying your NFT collection using Metaplex Candy Machine v3 on Solana. It assumes basic familiarity with command line, Node.js, and Solana wallets. Where possible, commands and config examples are provided.

## 1. Prerequisites

- Install Node.js ( $\geq 16$ ) and npm or yarn.
- Install Solana CLI: <https://docs.solana.com/cli/install-solana-cli-tools>
- Have a funded Solana wallet (CLI or Phantom).
- Install Metaplex JS tooling (Candy Machine v3) and `@metaplex-foundation/cli` or follow Metaplex docs.
- Have Arweave or IPFS account for hosting metadata & images.

## 2. Folder Preparation

Use the upload-ready package generated earlier. Structure should look like: `/assets/images` `hoodie_0001.png` ... `/metadata` `1.json` `2.json` ... `candy-machine-config.json` (see example) Ensure metadata JSONs reference final CDN URLs AFTER upload.

## 3. Example candy-machine-config.json

```
{ "price": 1.0, "number": 2500, "symbol": "CMR", "seller_fee_basis_points": 500, "isMutable": true,
  "retainAuthority": true, "goLiveDate": "2025-12-15T12:00:00Z", "creators": [ { "address": "", "share": 100
} ] }
```

## 4. Upload Images & Metadata

Option A — Arweave (recommended permanence): 1. Use arweave uploader (metaplex or arweave CLI) to upload `/assets/images` and `/assets/metadata`. 2. Capture the base URI and ensure metadata 'image' fields point to full URLs. Option B — IPFS (e.g., nft.storage or Pinata): 1. Upload images, pin them, get CIDs. 2. Update metadata 'image' fields to `https://ipfs.io/ipfs/hoodie_0001.png` etc.

## 5. Verify Metadata

Before minting, verify a few metadata files manually: • Fields: name, symbol, description, image (absolute URL), attributes array, properties.files.uri • Ensure JSON is valid (use JSONLint) and metadata matches marketplace expectations.

## 6. Deploy Candy Machine v3

1. Install Metaplex CLI or use the SDK. 2. Create a wallet and fund it with enough SOL for storage + fees. 3. Run the create and upload steps (example using metaplex CLI): `metaplex upload ./assets --config candy-machine-config.json --env mainnet-beta --keypair ~/.config/solana/devnet.json` 4. Create

the Candy Machine: `metaplex create_candy_machine --env mainnet-beta --keypair ~/.config/solana/devnet.json` 5. Set up guards, gatekeeping, or allowlist as needed. 6. Start the mint when ready. Refer to Metaplex docs for exact flags and options.

## 7. Post-Deployment Checklist

- Test minting with a small amount (one transaction).
- Verify metadata served from the final CDN on marketplaces.
- Monitor on-chain events and confirm creators' shares & royalties.
- Announce mint & provide support channels.

## 8. Troubleshooting & Tips

- If uploads fail due to size, batch them in smaller groups.
- Use a dedicated upload machine with stable network.
- Double-check `goLiveDate` and `timezone`.
- Keep backups of images & metadata before upload.
- For large teams, maintain a versioned folder (e.g., `/assets_v1`) and use git for templates and CSVs.

## 9. Useful Commands (Examples)

```
# check solana balance solana balance # create a keypair solana-keygen new --outfile
~/.config/solana/devnet.json # fund devnet wallet (for testing) solana airdrop 2 --url
https://api.devnet.solana.com # upload assets (example via metaplex) metaplex upload ./assets
--config candy-machine-config.json --env devnet --keypair ~/.config/solana/devnet.json
```

## 10. Team Handover Checklist

- Ensure README in repo is up-to-date.
- Confirm wallet addresses & creator shares.
- Provide one responsible operator for uploads.
- Share contact and escalation for deploy day.

## **Appendix A — Example Asset Naming Convention**

Images: hoodie\_0001.png, hoodie\_0002.png, ... Metadata: 1.json, 2.json, ... Template: templates/hoodie\_metadata\_template.json Traits: hoodie\_traits.csv