```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.21;

//Sets Contract up to use or copy imported contracts

import "@openzeppelin/contracts@4.9/token/ERC721/extensions/
ERC721URIStorage.sol";
import "@openzeppelin/contracts@4.9/access/Ownable.sol";

// Constructor tells MyContract to inherit both ERC721URIStorage "which inherits
from ERC721" &Ownable's functions
contract MyContract is ERC721URIStorage, Ownable {
  constructor() ERC721("MyContract", "MCNFT") {}

    //Public can to see the baseURI
  function _baseURI() internal pure override returns (string memory) {
    return "https://YOURIPFSURIORURL";
  }

  // Define the total supply of tokens
  uint256 public constant MAX_SUPPLY = 100;

  // Track the number of minted tokens
  uint256 public mintedTokens;

// Only the Deployer can mint
  function safeMint(address to, uint256 tokenId) public onlyOwner {
    // Check if the total supply is reached
    require(mintedTokens < MAX_SUPPLY, "Max supply of tokens reached");

    // Existing checks for recipient owning a token and tokenId validity

    _safeMint(to, tokenId);
    mintedTokens++;
  }


}
```