



niceHASH OS

beta

User Guide

# Table of contents

<b>NiceHash OS</b>	<b>3</b>
Quick setup guide	
Detailed setup guide	
Prerequisites	
Concepts	
Creating NiceHash OS flash drive	
Finish	
<b>How to create NiceHash OS bootable flash drive</b>	<b>4</b>
Prerequisites	
Install software for writing to USB flash drive	
Preparing NiceHash OS Flash drive	
<b>How to configure NiceHash OS</b>	<b>6</b>
What you will need	
NiceHash OS configuration	
Rig	
Access	
Network	
<b>NiceHash OS local access</b>	<b>9</b>
Prerequisites	
How to log in to the system	
Places to check out	
System configuration	
System logs	
System usage	
<b>Manual OC settings (advanced)</b>	<b>10</b>
TDP (valid for both NVIDIA and AMD)	
Core_clocks & memory_clocks (NVIDIA)	
Core_clocks & memory_clocks (AMD)	
<b>Algorithm IDs</b>	<b>11</b>
<b>NiceHash OS bug reporting</b>	<b>12</b>
Prerequisites	
Bug Reporting	

# NiceHash OS

## QUICK SETUP GUIDE

Below is a list of steps get NHOS up and running:

- **Download NiceHash OS** image from our website.
- **Flash downloaded NiceHash OS** image to the **USB flash drive**.
- **Update NiceHash OS configuration file** with your data.

## DETAILED SETUP GUIDE

### Prerequisites

- Your mining rig needs at least **2 GB** of **RAM**.

### Concepts

NiceHash OS is an **operating system** which loads from **USB flash drive** and **runs from computer memory**. It contains **all the tools and drivers needed** to bring your mining machine to life.

NiceHash OS flash drive is divided into two partitions, **SYSTEM** and **NHOS**. **SYSTEM** partition contains bootloader, Linux kernel and RAM file system, while **NHOS** partition contains NiceHash OS configuration file.

### Creating NiceHash OS flash drive

There are **two steps required** to create a **fully functional NiceHash OS flash drive**:

- 1.) **Create a USB flash drive containing an operating system**. Detailed instructions are available on *How to create NiceHash OS flash drive* page 4.
- 2.) **Update NiceHash OS configuration file with your data**. Detailed instructions are available on *How to configure NiceHash OS* page 6.

### Note

*This is a guide on how to prepare one NiceHash OS flash drive. If you plan to use the same setup on multiple machines then you can do the following. Go through the steps mentioned above for each NiceHash OS flash drives you need. After this, you will have multiple NiceHash OS flash drives which contain an invalid (empty) configuration. Now use valid configuration from your initial NiceHash OS flash drive and copy it to all newly created NiceHash OS drives.*

## FINISH

At this point, you should have everything prepared to start using NiceHash OS. All you need to do now is to **plug in NiceHash OS flash drive into your mining machine and turn it on!**

### Note

*Information on how to access mining machine locally is available on *NiceHash OS local access* page.*

### Note

*Information on how to report bugs is available on *NiceHash OS bug reporting* page.*

# How to create NiceHash OS bootable flash drive

This guide provides information on how to create bootable flash drive with NiceHash OS. Following the instructions below, you will download **NiceHash OS image** (an .img.gz file), and then use **special application that writes an image to a USB drive**. You cannot simply copy the image file to a USB drive, you must use special software to write it properly.

## PREREQUISITES

- A computer connected to the internet with a **Windows, macOS** or **Linux operating system**.
- USB flash drive with at least **2 GB of space**.
- You have a **special software to write image file to USB flash drive**.

### Note

For creating NiceHash OS flash drive, you will need elevated privileges on your system (administrator privileges on Windows and root privileges on macOS or Linux).

### Warning

The process described below will delete any data currently on your flash drive. Make sure to back up your flash drive's data to another storage location before proceeding.

## INSTALL SOFTWARE FOR WRITING TO USB FLASH DRIVE

There are many tools available for writing image files to USB drive. NiceHash is providing specially crafted **NiceHash Flash Tool** just for preparing bootable NiceHash OS flash drive(s). If you prefer to use something else there are a number of well established tools like **Etcher**, easy to use cross-platform image burning tool.

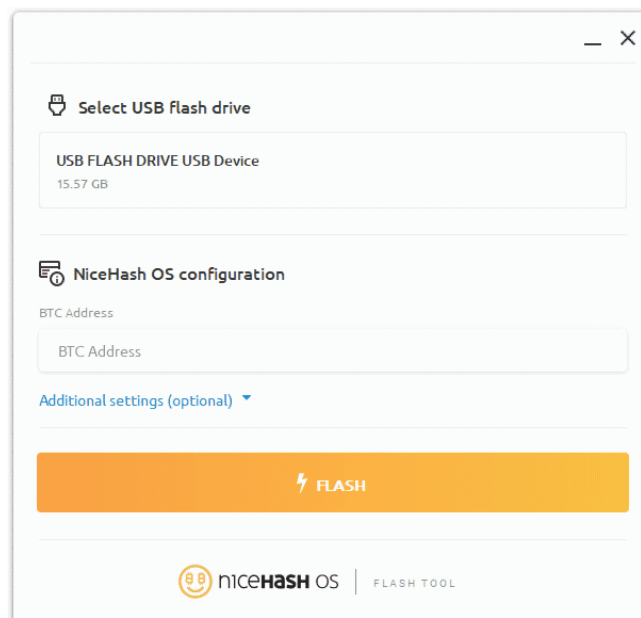
### Note

On systems like macOS and Linux users can also use command line tool `dd` to write image files to USB drive.

## PREPARING NICEHASH OS FLASH DRIVE

### 1. Using NiceHash Flash Tool (preferred)

**NiceHash Flash Tool** tool not only provides cross-platform image burning capabilities, but it is also extremely easy to use. There is no need to download NiceHash OS image beforehand, this tool will do this for you.



## 2. Using other tools

When using tools other than **NiceHash Flash Tool**, you need to download **the latest NiceHash OS image** from our website beforehand and then write it to USB drive using aforementioned Etcher or any other similar tool.

### **Note**

If you prefer to use command line tools instead, there are three simple steps to write an image file using dd tool

1. Decompress NiceHash OS image file

**gunzip nhos-x.x.x.img.gz**

2. Define output device to write an image file to

**/dev/sda**

3. Start image writing process

**dd if=nhos-x.x.x.img of=/dev/sda bs=4M && sync**

### **⚠ Warning**

Be careful defining output device, wrong usage of **dd** tool can lead to irreversible damage to your system!

# How to configure NiceHash OS

This guide provides information on how to correctly configure NiceHash OS to work once a flash drive has been created. If you do not have NiceHash OS flash drive created yet then jump to *How to create NiceHash OS flash drive* page for further instructions.

## WHAT YOU WILL NEED

Here is the list of preconditions that must be met before you can proceed:

- Computer with one of the following operating system: **Windows, MacOS** or **Linux**.
- **NiceHash OS flash drive**.
- **Text editor** you are familiar with.

### Note

Any text editor is fine as long as it does not modify the file format being edited. Depending on the operating system, you can use the native text editor (Notepad for Windows, TextEdit for MacOS, Vi for Linux), or any of the 3rd party editors listed below:

	Atom	Brackets	Visual Studio Code	Notepad++	Nano	Geany	KWrite
Windows	x	x	x	x			
Mac OS	x	x	x				
Linux	x	x	x		x	x	x

## NICEHASH OS CONFIGURATION

NiceHash OS flash drive is divided into two partitions, **SYSTEM** and **NHOS**, where NHOS partition contains the configuration data for the system to operate correctly.

When NiceHash OS flash drive is inserted into your computer, the system should **automatically detect NHOS partition** and **show it in system file manager** (Windows Explorer, macOS Finder, Linux Nautilus, etc.) as a disk on Windows or as a mounted device on macOS and Linux.

Using your file manager, navigate to the **location of the NHOS disk or mount point**. There you will find a **single configuration file** named `configuration.txt`.

To modify NiceHash OS configuration, open this configuration file with your preferred file editor where you will see the following file content structure:

```

{
  "rig": {
    "btc": "",
    "worker": "",
    "group": ""
  },
  "access": {
    "ssh": {
      "key": "" }
    },
    "network": {
      "wireless": {
        "ssid": "",
        "key": ""
      }
    }
  }
}

```

Configuration is divided into different sections: **rig**, **access**, and **network**. All the sections contain configuration entries which are named to be as self-explanatory as possible.

**BTC address** is the only **mandatory information** that needs to be present in the configuration and is part of the *rig* section, while *access* and *network* sections can be left *empty* or *removed* from the configuration file if not needed.

## Rig

This section is used to configure NiceHash OS mining feature.

**btc** - Your NiceHash BTC address. **[mandatory]**

**worker** - Name of your mining machine (sometimes referred to as **rig name**). Have in mind, that **worker** name is limited to alphanumeric characters from English alphabet, **a to z**, **A to Z**, **0 to 9** and characters - (hyphen), **\_**(underscore). **[optional]**

**group** - Put your mining machine into group. **[optional]**

## Access

This section is used to **configure mining machine network settings**. There is only wireless configuration entry here, and if you are using wireless infrastructure, you must fill in this section, otherwise leave it empty or remove from configuration file.

**ssh** - SSH public key for user authentication. **[optional]**

## Note

*There is no special configuration for local shell user access. More information on how to access mining machine locally is available on NiceHash OS local access page.*

## Network

This section is used to **configure mining machine network settings**. There is only wireless configuration entry here, and if you are using wireless infrastructure, you must fill in this section, otherwise leave it empty or remove from configuration file.

**ssid** - SSID to provide the name for a wireless network access. **[optional]**

**key** - Security key for wireless network access authentication. **[optional]**

### Note

*For security reasons only **WPA** and **WPA2** security protocols are supported.*

### Note

*There are no configuration entries to configure wired network setting. Also there are no configuration entries to define static IP address to be assigned to the mining machine. No matter if you are running mining machine on a wired or wireless infrastructure, IP address is always assigned dynamically using DHCP protocol.*

### Example on how configuration might look like after modification.

```
{
  "rig": {
    "btc": "2N8xDN798uKMgPxTt35pgmGcdpJnSAvgsMF",
    "worker": "centaurus",
    "group": "constellation"
  },
  "access": {
    "ssh": {
      "key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCoWwJIm9JNgWzPrsMAeYWdM4nAkCET4j1kONsGPE2GeKul/
4dTiq8X8aTKVdLLXOTQxBsOjb6J4umgVioTuorthjD0lYM3HDp55BnBgcnXXm7TfKzWKyCcbXvpOZA1pdzLKT
o8bSBWjq4P2J0xPO6A6QHqVqs2LDPC5SyDMYrXOKrPLHfNxxzg9mvry49RtQJSzBICnBWDC28pNSCjvKbeHzE
A85Quy4ctR7A7cHHeR0G3k/Xozdc8/
eUptxhbW2M4t4uUg4Tnh4OQEPJKQ5j4zvKqRrxzMV1Kvxuarxbouwci569ulaOYDUQIOs8BB57d5IP3HRvsG4
Ok8HosIxTJ"
    }
  },
  "network": {
    "wireless": {
      "ssid": "zodiac",
      "key": "8!Lf@I5s3tpY"
    }
  }
}
```

When you are done with configuration modifications, **save your changes** and **safely unplug NiceHash OS flash drive** from your computer.



# NiceHash OS local access

This guide provides information on how to **access NiceHash OS locally**.

## PREREQUISITES

- Access to the **mining machine** with a **running NiceHash OS system**
- **Keyboard** and **monitor connected** to the **mining machine**.

## HOW TO LOG IN TO THE SYSTEM

On system start, you are presented with a screen where you can **enter access credentials to log in** to the system. There is only **one user configured for accessing NiceHash OS** and there is **no password required** to log in locally. Just **enter username nhos** and **hit Enter key** (Return on some systems).

**NiceHash OS is a minimal Linux operating system** based on *Tiny Core Linux*. If you are familiar with any Linux distribution, then you will feel comfortable using NiceHash OS as well.

## PLACES TO CHECK OUT

### System configuration

NiceHash OS configuration prepared while creating NiceHash OS flash drive is located under `/mnt/nhos` directory in `configuration.txt` file.

### System logs

System startup and application logs are located under `/var/log/nhos` directory.

**nhos\_boot.log** - contains information on NiceHash OS boot sequence

**nhos\_nhm.log** - contains information on NiceHash Miner operations

**nhos\_update.log** - contains information on NiceHash OS update service operations

## SYSTEM USAGE

There are **no special usage cases logging in to the NiceHash OS system** other than **checking system configuration** and **logs** in case something is not working as expected. Please remember that NiceHash OS is **running completely from computer memory** and **any changes** you might have done to the system are **lost on reboot**. The only location where changes are persisted are the ones under `/mnt/nhos` directory.

# Manual OC settings (advanced)

You can apply overclocking (OC) settings by manually changing the parameters in the `device_settings.json` file.

**device\_settings.json** is generated on your NHOS bootable USB drive after the first successful boot on your mining rig. After that you can open `device_settings.json` file and change power (TDP) settings, core and memory clocks for every device and algorithm separately.

```
tdp:                "default"  
core_clocks:        "default"  
memory_clocks:      "default"
```

## TDP (VALID FOR BOTH NVIDIA AND AMD)

TDP is set as a percentage parameter for both NVIDIA and AMD graphics cards. If you want to change TDP you have to change the "default" value.

For example, if you want to set TDP to 85% then you have to change this value to "tdp": 85.

For example, if you want to set TDP to 110%, then you have to change this value to "tdp": 110.

## Core\_clocks & memory\_clocks (NVIDIA)

If you want to change core and memory clocks for NVIDIA graphics cards you have to change the "default" value. Value is relative to card default value.

For example, if you want to raise your core or memory clock for 50 Hz then you have to change this value to "core\_clocks": 50 or "memory\_clocks": 50.

For example, if you want to lower your core or memory clock for 50 Hz then you have to change this value to "core\_clocks": -50 or "memory\_clocks": -50.

## Core\_clocks & memory\_clocks (AMD)

If you want to change core and memory clocks for AMD graphics cards you have to change the "default" value. Values for AMD cards are absolute.

For example, if you want to raise your core or memory clock from 1800 to 1850 Hz then you have change this value to "core\_clocks": 1850 or "memory\_clocks": 1850.

For example, if you want to lower your core or memory clock from 1800 to 1750 Hz then you have change this value to "core\_clocks": 1750 or "memory\_clocks": 1750.

## ALGORITHM IDs

In the table below you can find IDs for every algorithm

Algorithm	ID
Scrypt	0
SHA256	1
ScryptNf	2
X11	3
X13	4
Keccak	5
X15	6
Nist5	7
NeoScrypt	8
Lyra2RE	9
WhirpoolX	10
Qubit	11
Quark	12
Axiom	13
Lyra2REv2	14
ScryptJaneNf16	15
Blake256r8	16
Blake256r14	17
Blake256r8vnl	18
Hodl	19
DaggerHashimoto	20
Decred	21
CryptoNight	22

Algorithm	ID
Lbry	23
Equihash	24
Pascal	25
X11Gost	26
Sia	27
Blake2s	28
Skunk	29
CryptoNightV7	30
CryptoNightHeavy	31
Lyra2Z	32
X16R	33
CryptoNightV8	34
SHA256AsicBoost	35
Zhash	36
Beam	37
GrinCuckaroo29	38
GrinCuckatoo31	39
Lyra2REv3	40
CryptoNightR	42
CuckoCycle	43
GrinCuckarood29	44
Beamv2	45
X16Rv2	46

# NiceHash OS bug reporting

This guide provides information on how to **report a bug** concerning NiceHash OS.

## PREREQUISITES

- Access to the **mining machine** with a **running NiceHash OS system**
- You are **familiar with a Linux shell** and how to access your mining machine locally or remotely.

## BUG REPORTING

If you believe that you've found a bug that should be reported to NiceHash, then make sure to **describe it as precise as possible**. But be sure to also send us a **copy of the NiceHash OS system dump** file, as this will be of great help to our developers.



There are two commands available which you can use to generate NiceHash OS system dump file.

`nhos_system_dump_save` - will generate **system dump file** and **save it to the USB flash drive**.

`nhos_system_dump_push` - will generate **system dump file** and **push it to the NiceHash network storage**.

### **Note**

*If you are familiar with the Linux operating system then it should be fairly simple to retrieve system dump file from your mining machine. You can access your mining machine locally and then just copy the file to an external media or access your mining machine remotely using SFTP client (WinSCP, FileZilla, CyberDuck, etc.) and get the file this way.*



niceHASH OS  
beta