Dedicated server setup

	This article may need cleanup to meet quality standards. Please help improve this if you can. The talk page may contain suggestions.

This page discusses the installation and configuration of a dedicated ARK server.

For scripts that you can run on your own dedicated server, see Dedicated server scripts.

Prerequisites

Your OS **must** be a 64-bit OS (ShooterGameServer is a 64-bit executable and as such will not run on a 32-bit install of Ubuntu or Windows). The server has been tested on Ubuntu 14.04 and should also support any Windows host where the prerequisites found in `_CommonRedist` can be installed.

Hardware

The server requires at least 6GB of RAM to start. Memory requirements increase as the number of connected players increases. This is also dependant on your settings/mods and configurable options. Some mods increase RAM more than others. On Windows Server 2012 R2, the server uses 2.3GB of RAM without any clients. On Linux, the installation uses 3.2GB of RAM without any clients.

Network

The server listens for incoming connections on the ports listed below. Ensure your network configuration allows incoming connections to these ports and directs them to the host that will be running your dedicated server.

Port	Purpose
UDP 27015	Query port for Steam's server browser
UDP 7777	Game client port
UDP 7778	Raw UDP socket port (always Game client port +1)
TCP 27020	RCON for remote console server access (optional)

You can host multiple dedicated servers on the same host, however be aware that CPU and memory should be monitored carefully if they are busy servers rather than test servers. For example:

Server instance	Game port	Raw UDP port	Query port	RCON port
Server game instance 1	7777	7778	27015	27020
Server game instance 2	7779	7780	27016	27021

Server game instance 3	7781	7782	27017	27022
Server game instance 4 completely different ports	9999	10000	37015	37016

> **Note:** Query Port cannot be between 27020 and 27050 due to Steam using those ports.

Operating system

Windows

The server should run on most modern systems. If you have information on running it on a specific version of Windows, please update this section.

Historically the server required Microsoft Visual C++ 2013 Redistributable and a certain revision of the DirectX redistributables (at least June 2010) which were included, however the installations are no longer included and most DLLs are already included.

Linux

64bit Linux servers will require 32 bit binaries to install/run steamcmd.

sudo apt-get install lib32gcc1

The game server requires *glibc* 2.14 or greater. Ubuntu 14.04 (and newer) and Debian 8 (Jessie) satisfy this requirement automatically, but older versions, such as Debian 7 (Wheezy), do not. To install the required version of *glibc* on a Debian 7 (Wheezy) host:

Add the following lines to `/etc/apt/sources.list`:
Experimental/unstable (sid) repositories
deb http://ftp.debian.org/debian experimental main
deb http://ftp.debian.org/debian sid main

To update the host's available packages list and install the updated *glibc* library package, run the following commands via *sudo* or a root shell:
apt-get update
apt-get -t experimental install libc6-dev

> **Note:** This procedure does *not* upgrade the host to Debian Unstable (sid); only *glibc* itself and any packages it depends upon are updated.

For other older distributions that don't include *glibc* 2.14 or newer, refer to the distribution's documentation and support forums for guidance.

Open Files Limit

> **Note:** *This section doesn't apply if you'll be using* systemd *to launch the dedicated server (as described below in [#Automatic Startup](#)), as it can set this limit at runtime.*

To ensure that the host's open files limit is high enough to support the game server:

Add the following lines to `/etc/sysctl.conf`:
fs.file-max=100000

then run the following command via *sudo* or a root shell to apply the change:

$ sysctl -p /etc/sysctl.conf

Add the following lines to `/etc/security/limits.conf`:
* soft nofile 1000000
* hard nofile 1000000

Add the following line to `/etc/pam.d/common-session`:
session required pam_limits.so

> **Warning:** Without these changes, the game server may not successfully launch. If the server appears to start, but uses a high amount of CPU time without using at least 5.5GB of RAM, it hasn't been able to open all the files it needs and the above change should be applied.

Tuned is a daemon that monitors the use of system components and dynamically tunes system settings, currently only on RHEL7, CentOS7.x and Fedora.

Note: You need to set tuned to „throughput-performance“. Otherwise all changes in /etc/security/limits.conf will ignored!

SteamCMD

The dedicated server is available for both Linux and Windows platforms. For both platforms, SteamCMD is used to download the server files. Refer to that page for detailed instructions on its installation and usage; the instructions included in the steps below are deliberately concise for space considerations.

Remember that you can use the `force_install_dir <path>` SteamCMD command to change the default installation location.

Server Installation

Install SteamCMD on your host.

Create a folder to house the server files on a volume with at least 25GB of free disk space. The server files consume more than 17GB of space, but it's wise to plan for saves and future content updates which will consume additional space.

Launch **SteamCMD** on your host and use it to download the server files. (Remove the < > and replace them with double quotes, for example, `force_install_dir "C:\Program Files (x86)\Ark Server"`) Use app id `376030` for **Survival Evolved** or use `445400` for **Survival of The Fittest**.
Steam> login anonymous
Steam> force_install_dir <install_dir>
Steam> app_update 376030 validate
Steam> exit

Or, run from command line:
steamcmd +login anonymous +force_install_dir <install_dir> +app_update 376030 +quit

Replace `<install_dir>` with the full path to the folder created in Step 2. On Linux, this will be a path like `/home/steam/servers/ark`. On Windows, this will be a path like `C:\arkserver`.

Create a script to launch the server with your desired options and settings. For simplicity's sake, place it in the same folder where the server files were downloaded. On Windows, this is a batch file; on Linux, this is a shell script. Examples are provided below. For both platforms, the server options are specified with the same basic syntax.
For Windows, create `start_server.bat` and place it in **YOUR_ARK_SERVER_FOLDER**/ShooterGame /Binaries/Win64/ (remove the '<' '>' symbols!)
ShooterGameServer.exe TheIsland?listen?SessionName=<server_name>?ServerPassword= <join_password>?ServerAdminPassword=<admin_password>?Port=<port>?QueryPort= <query_port>?MaxPlayers=<max_players>

For Linux, create `server_start.sh` and place it in **YOUR_ARK_SERVER_FOLDER**/ShooterGame /Binaries/Linux/ (remove the '<' '>' symbols!)
#! /bin/bash
./ShooterGameServer TheIsland?listen?SessionName=<server_name>?ServerPassword= <join_password>?ServerAdminPassword=<admin_password> -server -log

After creating the script, make it executable:
$ chmod +x server_start.sh

In both of these examples, replace `<server_name>` with the desired name for your server, `<join_password>` with whatever password players must provide to join your server and `<admin_password>` with the password that must be provided to gain administrator access to the server. If

no player join password is desired, remove the entire option from the list (including the `?ServerPassword=` parameter itself).

Finally, to launch the server, run the script created in the previous step.

Port Forwarding and Firewall

For your server to become visible in both the Ark server lists and the Steam lists, do the following:

Windows (Firewall and Allow Rules)

If you're on Windows 10, click on Start and search for "wf.msc" without the quotes. Click the "Run as administrator" option.

Click "Inbound Rules" on the left panel, then click "New Rule..." on the right panel.

When the New Inbound Rule Wizard opens up, select Port, then Next.

The wizard will present options for choosing the protocol and specifying ports. Leave the default selection of the TCP protocol and "Specific local ports". List the ports you plan to forward into the box (default 27020 for RCON). Use a comma to separate multiple ports. Select Next to continue.

Select "Allow the connection" and continue with the wizard.

Make sure all of the boxes are selected for where the rule applies.

Enter a name (required) and a description (optional). It is recommended to append the protocol (TCP or UDP) to the rule name.

Repeat these steps, but choose UDP on the "Protocol and Ports" page. Make sure to enter your steam browser/query port (27015 default) and game port (7777 default). You also need the raw udp socket port (7778 default, always your game port +1) if using ?bRawSockets.

When finished, you should have a rule for TCP and a rule for UDP in the Inbound Rules section.

This section and the port forwarding section go hand in hand.

Close the advanced firewall window and open Windows Firewall again.

Click "Allow an app or feature through Windows Firewall". Scroll down to ShooterGame. Make sure all apps named ShooterGame have the Private and Public boxes checked.

You're done. If your ports don't work when you forward them, you can edit your Firewall inbound rules by going back to the advanced tab, selecting the Inbound Rules tab, right-clicking the rules you created, then clicking properties.

OS X (firewall application)

Input needed.

Linux

For most Linux distros, if there is a firewall at all it will be iptables based. Most modern distributions will use a different interface, it's a good idea to use that instead of direct iptables commands.

All firewall changes must be made either as root or with sudo. These open ports 7777 7778 and 27015 for UDP and optionally 27020 for TCP. You may want to adjust the list of ports you need by editing the "for port in..." line.

UFW

This script will open the ports in a way that will persist across reboots. UFW is the firewall on Ubuntu

```
#!/bin/sh
if [[ $EUID -ne 0 ]]; then
   echo "This must be run as root"
   exit 1
fi
for port in 7777 7778 27015; do
   ufw allow $port/udp
done
#Uncomment the next if you want to open the default rcon port
#ufw allow 27020/tcp
```

FirewallD

This script will open the ports in a way that will persist across reboots. FirewallD is the firewall on RHEL7, CentOS7.x and Fedora.

```
#!/bin/sh
if [[ $EUID -ne 0 ]]; then
   echo "This must be run as root"
   exit 1
fi
for port in 7777 7778 27015; do
   firewall-cmd --permanent --add-port $port/udp
done
#Uncomment the next if you want to open the default rcon port
```

```
#firewall-cmd --permanent --add-port 27020/tcp
firewall-cmd --reload
```

iptables

Keep in mind that this is only a temporary solution and should only be used if neither of the other scripts apply for your system. To make it safe for reboots see the documentation of your distribution, how to add firewall ports to the existing configuration files - or how to add scripts to the boot process.

```
#!/bin/sh
if [[ $EUID -ne 0 ]]; then
   echo "This must be run as root"
   exit 1
fi
for port in 7777 7778 27015; do
  iptables -t filter -I INPUT -p udp --dport $port -j ACCEPT
done
#Uncomment the next if you want to open the default rcon port
#iptables -t filter -I INPUT -p tcp --dport 27020 -j ACCEPT
```

Port Forwarding

There are many websites with tutorials on port forwarding. It is highly suggested you visit portforward.com, which offers tutorials for the majority of routers.

What ports work depends on you. Try forwarding 27015 as the port and 27016 as the query port. 27015, 27016 and 7777 are the most common ports that work.

You **must** forward both TCP and UDP for your server to show up on any lists. You can make sure your ports are open by using the tool from portforward.com, that is, after you enter the ports that you have forwarded in the start_server.bat file you should have created earlier.

For example, start `ShooterGameServer.exe TheIsland?listen?SessionName=<server_name>?ServerPassword=<join_password>?ServerAdminPassword=<admin_password>?Port=7777?QueryPort=27015?MaxPlayers=<max_players>`

Good luck, port forwarding is not very easy.

Finding Your Server in Lists

You can access your server via the Steam servers list or the Ark unofficial list. Just search for your name in the Ark list, or find your IP address in the Steam server list. It is best to add your servers IP:PORT to your Steam

favorites and joining using the 'favorites' tab in game. The in game list has a limit on the servers it will show at one time. There is no guarantee your server will appear in the unofficial lists all the time.

Good luck!

Automatic Startup

Windows (via Scheduled Task)

You can configure a scheduled task to automatically run a batch file and start the dedicated server when the system boots.

Open 'Task Scheduler'

Create Basic Task

The 'Create Basic Task Wizard' will appear, name the task whatever you want

Set the Trigger to 'When the computer starts'

Set the Action to 'Start a program'.

You will need to browse to the program/script you wish to start (which is your .BAT batch file that starts your server).

Depending on how your batch file works, you may need to set the "Start in (optional)" path, so the working directory of your batch file is correct. (If your batch file simply contains the "start ShooterGame ..." command, then you should set the "Start in (optional)" path of your scheduled task to the directory that contains the batch file)

Before you create the scheduled task, click the checkbox to "open properties window" after it's created

In the General Tab under the Security Options frame, you probably want to choose "Run whether user is logged on or not"

I personally checked "run with highest privileges" just in case. This might not be necessary.

Linux (via *systemd*)

> **Note:** systemd is not available for Ubuntu 14.04.

As an alternative to using a script to launch the dedicated server manually, hosts running *systemd* can be configured to automatically start the dedicated server when the system boots. When using this method to manage the server, using `GameUserSettings.ini` to specify its settings is highly recommended. Refer to Admin Game Commands for more information.

Create a file named `/etc/systemd/system/ark-dedicated.service` with the following contents:

```
[Unit]
Description=ARK: Survival Evolved dedicated server
Wants=network-online.target
After=syslog.target network.target nss-lookup.target network-online.target

[Service]
ExecStartPre=/home/steam/steamcmd +login anonymous +force_install_dir /home/steam/servers/ark
+app_update 376030 +quit
ExecStart=/home/steam/servers/ark/ShooterGame/Binaries/Linux/ShooterGameServer
TheIsland?listen?SessionName=<session_name> -server -log
WorkingDirectory=/home/steam/servers/ark/ShooterGame/Binaries/Linux
LimitNOFILE=100000
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s INT $MAINPID
User=steam
Group=steam

[Install]
WantedBy=multi-user.target
```

The `ExecStart` line specifies the command to run in order to start the service. It uses the same syntax as the startup examples shown above (in #Server Installation), with all the same arguments, but here it's important to adjust it to reflect the location of your ARK dedicated server on the host. Use the **full** path to the server executable, as shown above.

Be sure to replace `<SESSION_NAME>` with the desired session name for your server as well.

Finally, adjust the `User` and `Group` settings for your host. Without these two options in `ark-dedicated.service`, the dedicated server will be run as the root user. ***This is unsafe,*** *as any vulnerability in the dedicated server could result in an attacker gaining remote superuser access on the host.*

It's recommended to run the dedicated server in an unprivileged account used solely for this purpose. In the example above, the user account "steam" is used. It's a member of the "steam" group, which is generally created along with the account.

WorkingDirectory option is required to fix some issues. (For example not having **WorkingDirectory** set will fail to download mods (if using with -automanagedmods))

To install systemd:

```
apt-get install systemd systemd-sysv
```

You will need to reboot after installation to initialize systemd.

Activate the new service so it starts automatically when the host boots by running the following command via *sudo* or a root shell:

systemctl enable ark-dedicated

After running this command, the dedicated server will automatically start when the host does. The command doesn't start the server immediately, so to launch the dedicated server after enabling it, run:

systemctl start ark-dedicated

After following these steps, your server should be up and running and be automatically managed by the host on startup and shutdown.

Post-Setup Management

The server can be stopped by running:

systemctl stop ark-dedicated

and its current status can be viewed (whether it's running or not) by running:

systemctl status ark-dedicated

> **Note:** If you need to update the `ark-dedicated.service` file (to change the dedicated server's startup options or to adjust its path), run the following command to ensure your changes are applied:

systemctl daemon-reload

Updating

To update the server when a new version is released, repeat the same *SteamCMD* commands shown in the previous section. Be sure to use the correct set of commands for your platform. Refer to *SteamCMD's* documentation for details on automating this process.

If you use the systemd startup file you can easily integrate the update-mechanism to be run before every start of the game.

SteamCMD provides additional tools to make installation and updates easier and more seamless.

Console Commands

While running the game, the command console can be accessed with either the `[~]` (tilde) or `[TAB]` keys (depending on game version, default configuration and your keyboard layout). Once in the console, to activate

administrator commands, enter:

enablecheats <admin_password>

Replace **`<admin_password>`** with the server's administrator password.

Refer to Admin Game Commands for a list of available commands.

If the console can't be opened in-game, exit the game, open the game's `DefaultInput.ini` file (located in your Steam library in the folder named `steamapps\common\ARK\ShooterGame\Config`) with a text editor and locate the line (near the bottom of the file) that reads:

;+ConsoleKeys=Tab

Remove the semicolon (;), changing the line to read:

+ConsoleKeys=Tab

Save the file and launch the game. The console should be accessible.

Backing Up Server Data

To make a backup of the server data, simply copy the folder named `ShooterGame/Saved` (and his content) to the desired backup location. This folder contains all tribe, player and world data for the server. Performing a backup is recommended before updating a server to a new release.

Troubleshooting

Server loses characters or world data upon restart

If your server isn't retaining characters or world data when it's restarted, it may not have full access to the folder where it's installed.

The first time the server is run, it should create a folder named `ShooterGame/Saved` containing various configuration and save files. If the server has run at least once but hasn't created this folder, it may not have been able to create new folders and files there. Verify that the folder where the server is installed is readable and writable by the user account that actually runs the server. (For Windows servers, check the status of User Account Control)

The server software, by default, is set-up to automatically save world data every 15 minutes. If your server crashes before the first 15 minutes are up, you will not have any data saved.

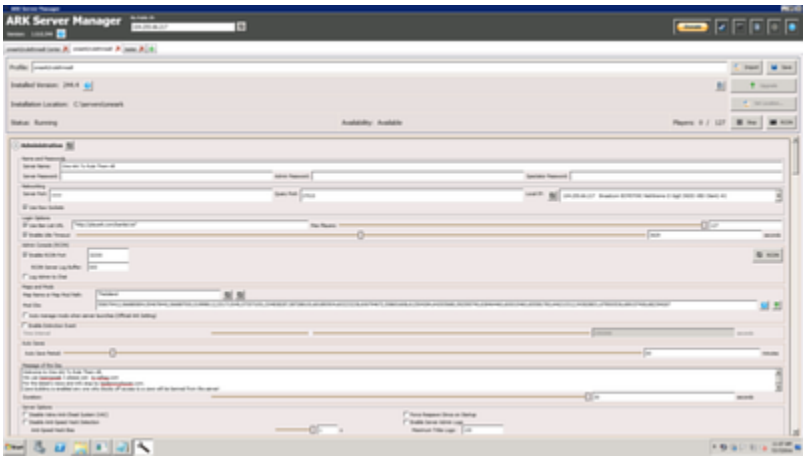
Linux server exhibits high CPU usage and low memory usage (Below 1GB)

This can occur when the server can't open all the files it needs due to an insufficient open files limit on the host.

Refer to the Linux section above and follow the procedure to increase the host's open files limit.

Tool - ARK Server Manager

You can also use this tool to manage your server when it is installed: https://steamcommunity.com/sharedfiles/filedetails/?id=468312476



The Ark Server Manager is designed to help you set up and maintain your own Ark: Survival Evolved (tm) dedicated servers. It provides a simple user interface allowing you to create and edit server profiles containing all of the settings you need to customize the playing experience for you and your friends.

What it does:

Manages the SteamCMD tool and Server installations/updates.

Organizes settings and writes INI files and command-line arguments automatically.

Tracks server status and allows direct control of server state.

Save and load server profiles - even import directly from an existing server deployment.

Scheduled updates server files and mods.

Works with Survival of the fittest.

Notes

If you're planning to have players join from the Epic Games version of ARK, you must have the `-crossplay` command line option set.

References

External links

This material was partially based on a Server Setup guide by ShuwA.

gameserversetup.com guide

survivalservers.com guide

http://steamcommunity.com/app/346110/discussions/0/615086038673139870/

hostnoc.com guide

bestarkhosting.com guide

comparegamehosting.com guide

topserver.network Ark Server Setup Guide