Unreal Tournament 4 Hub Setup Guide for 2024

By neckutter1 of the Agents of Oblivion clan

I received assistance from Zunnie (Rest in Peace), MII, and Skandalouz from UTCC, as well as wrobot (Rest in Peace), from Absolute Team. Additional information was gathered from numerous web tutorials, too numerous to individually acknowledge. Gratitude to everyone involved!

Please be aware that many commands assume you have utilized "cd .." or "cd /SOME_FOLDER/" to return to your user home directory, denoted as MY_USER_NAME. The placeholder password MY_PASSWORD is used in lieu of your actual username and password.

Let us assume /home/neckutter1/LinuxServer/ is the name of my server installation folder.

Example:
/home/neckutter1/LinuxServer/UnrealTournament/Saved/Config/WindowsServer/Engine.ini

Additionally, I use the terms "Hub" and "Server" interchangeably, aligning with common usage among game players. However, for clarity, it's important to specify that we'll be setting up a "Hub." Certain commands necessitate root access; if a command requires this, it will explicitly indicate so. To execute such commands, prepend "sudo" before the command, prompting you for your root password before execution. Typically, this is only necessary once per session. When copying and pasting commands, include or exclude the sudo command based on your specific circumstances.

Note: I am not a Linux expert. It is conceivable that at the time of writing or in the future, the system I've assembled could be susceptible to exploits or hacking. There may also be alternative methods or shortcuts for achieving the same results. This guide reflects the approach I used, and you should use it at your own risk!

Part 1: Centos 7 Setup

This outlines the process I followed to get Unreal Tournament 4 (also known as Unreal Tournament Pre Alpha) Hub running on an older machine with Centos 7.

To begin, select a suitable machine. The test machine in this case features an Atom processor, single core, 1.6 GHz, and 1 GB of memory. While the server does operate on this configuration, it is not considered ideal. In contrast, three fully operational hubs in my setup boast Dual Quad Xeons with 12 GB of memory each. You may want to aim for a system that falls between the extremes, leaning

more towards the higher specifications for optimal performance. The intended setup will be a headless one, meaning no monitor, mouse, or keyboard will be connected once it's up and running—just a power cord and an ethernet cable.

Before you start, access the computer's BIOS or UEFI settings and ensure that it can boot without a mouse and keyboard connected. Look for options like "Halt on no errors" or "Halt on all but keyboard." On my machine, this setting was on the first page of the BIOS setup. While in the BIOS, it's advisable to configure the machine to automatically power back on if it loses power. On my machine, this setting was located in the BIOS under the "power" tab.

Now, let's proceed to obtain the operating system. While I burned mine to a DVD, you can alternatively create a bootable USB drive for this purpose. You can find guidance on creating a bootable USB drive through online searches. For this tutorial, I'll be using a DVD.

I obtained Centos 7 from the following link:
http://archive.kernel.org/centosvault/centos/7.0.1406/isos/x86_64/CentOS-7.0-1406-x86_64-Minimal.iso

Next, I burned the ISO file to a DVD using the built-in utility in Windows 10. To boot the machine from the DVD, I pressed "ESC" at the "Compaq" logo on my machine, which allowed me to select the boot device. You may need to refer to your manufacturer's website or consult your machine's model on Google to determine how to boot from a DVD.

I chose "Install Centos 7" and pressed "enter." After a brief wait, I was presented with a screen prompting me to select my Language and keyboard. I opted for "English" for both selections.

Following that, I was prompted for installation details. Here are the selections I made (in this order):

Network & Hostname Configuration:

To initiate the process, I clicked the icon located at the top right and toggled it on. Moving to the bottom left, a box prompted me for my hostname. Since my machine's name was "Tarja," I cleared the existing information and entered "Tarja" into the box. Subsequently, I clicked on "Configure" at the bottom right, taking note of the connection name, which may be needed in the future—mine was labeled "enp2s0."

Moving to the IPv4 tab, I changed the "Method" from DHCP to Manual. Then, I clicked "Add" and configured the Address, Netmask, Gateway, and DNS server settings. For reference, I used the following values:

- IP Address: 192.168.1.99

- Netmask: 255.255.255.0

- Gateway: 192.168.1.1

- DNS Server: 8.8.8.8

After making these adjustments, I clicked "Done" at the bottom right of the dialogue box. Additionally, on my router, I set the DMZ to this computer's IP address. Alternatively, you can forward the necessary ports (which will be listed later) in the router settings.

Finally, I clicked "Done" at the top left to complete the configuration.

Date & Time Configuration:

Configuring date and time is a straightforward process through the graphical interface. First, I turned off "Network Time" at the top right. Then, I adjusted my time format from "24 hours" to "AM/PM" before turning "Network Time" back on. This change was made based on my personal preference.

Next, I used the mouse to select my time zone on the map; in my case, it was "Los Angeles." Once the time zone was set, I clicked "Done" at the top left to complete the configuration.

Installation Destination Configuration:

Now, it's time to select the location for installing the server Operating System. I opted to use the entire drive for my installation. To do this, I clicked on "Installation Destination" and selected the icon corresponding to my disk (a 160 GB disk in my case). It's crucial to note that the defaults in this section will wipe the entire disk and configure it automatically for the server. If you wish to explore alternative methods or use a different drive without erasing the entire disk, you may want to consult Google for guidance. This guide specifically focuses on creating a Centos 7-only system from scratch.

After selecting the disk, I clicked "Done" at the top left. I then clicked "Reclaim Space" from the opened dialogue box. Another box appeared, and I chose "Delete All," followed by "Reclaim Space" again, returning to the main menu. If your disk is brand new or has no partitions, you might not need to reclaim space; the system will automatically set it up for you.

Skipping other options, I clicked "Begin Installation" at the bottom right. The installation process commenced, allowing me to set a Root password and create a user account along with its password. Under "Configuration," I selected the two options. **While creating the user account, there was an option to make it an Administrator account, Click this!** I ensured that the "Require Password" box was checked, filled out the information, and clicked "Done" at the top left. Then, I waited for the installation to complete.

Prepare Centos 7 for Configuration:

Upon completion of the installation, the system prompted for a reboot, marking the start of the setup for the new Centos 7 server.
The MOST CRITICAL step after the reboot is to disable SSH login for root. To do this, log in as root and execute the following command:

vi /etc/ssh/sshd_config

use the arrow keys to find the line that says

#PermitRootLogin yes

press 'i' to enter edit mode.

change yes to say no and delete the #

Press 'esc' and then ':' and type in wq and press enter.

Press "Enter" to proceed.

After this, I shutdown my server:
shutdown -h now

and put it where it will now permanently live, headless, with only an ethernet cable and a power cable.

Centos 7 Configuration via copy and paste, and using putty:

All subsequent tasks will be executed through SSH. I will utilize the "Putty" program to remotely log in, make adjustments, and configure the machine. Once the configuration is complete, the machine will be located 22 miles away from my current location. For now, I will perform the operations via "Putty" from my Windows 10 machine.

You can download "Putty" for Windows https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit0.71-installer.msi

After installing Putty, I remotely logged in using the server's local IP address set earlier: 192.168.1.99. I verified that root access via SSH was indeed disabled. Subsequently, I logged in with my regular username. You can duplicate my process by copying the direct commands from this tutorial by copying and pasting these commands. Highlight a command from this tutorial, press "Ctrl + C," then left-click on the Putty screen, and right-click to paste).

If there arises a need to execute a command with elevated privileges, we will prefix the command with the "sudo" line and provide the password when prompted.

Example:

sudo yum update -y

Appending the "-y" at the end eliminates the need for manual confirmation, automatically consenting to execute the following commands.

When you finish a putty session, hold "Ctrl" and press "D" to log out, and Putty will close.

The next step is to ensure that the system is up to date.
The next thing we should do is make sure that we are up to date.:

sudo yum update -y

and then press “Enter”.

Then it showed me the progress as it downloaded and installed the updates.
Next, I did some setup for various other things that I would need:

Disable SELinux.
Install zip, unzip and wget.
Install the Apache webserver. While not mandatory, this can be highly beneficial, so I include it as a standard step.
Install newer version of python as well as some additional stuff needed for our UTCC redirection.
Opening some ports that need to be opened.

Disabling SELinux:

sudo vi /etc/selinux/config

Change “Enforcing” to “Disabled”
Reboot the system by typing:

sudo shutdown -r now or **reboot**

and then press “Enter”.

Check the status of SELinux when the system comes back online, by typing:

sudo sestatus

and then pressing enter.

Install zip, unzip, Apache and wget.

sudo yum install zip unzip httpd wget -y

Enable Apache to start at boot, and restart the service.

sudo systemctl enable httpd.service
then:
sudo systemctl restart httpd.service

Install newer version of Python
First type:

sudo yum install centos-release-scl -y

Then do:

sudo yum install rh-python36 -y

Every time you need to use Python you must first enable it by:

scl enable rh-python36 bash

You can verify that it is running by:

python --version

We will need to add some libraries and modules to Python, but we will do that later.

Install Samba:

sudo yum install samba -y

Now we will open some ports to the firewall. We also open some services first:

sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=samba

Ports:

sudo firewall-cmd --zone=public --permanent --add-port=8000-016/udp
sudo firewall-cmd --zone=public --permanent --add-port=7777/udp
sudo firewall-cmd --zone=public --permanent --add-port=7787/udp
sudo firewall-cmd --zone=public --permanent --add-port=6500/udp
sudo firewall-cmd --zone=public --permanent --add-port=13000/udp
sudo firewall-cmd --zone=public --permanent --add-port=14000/udp
sudo firewall-cmd --zone=public --permanent --add-port=15000/udp
sudo systemctl restart firewalld

Now it was time to configure the Samba server:

sudo systemctl enable smb
sudo systemctl enable nmb
sudo smbpasswd – User_Name

I also edited the configuration file:

sudo vi /etc/samba/smb.conf

I changed the workgroup from "samba" to "WORKGROUP"
Under homes, I changed "Browseable to "yes" I
added a line under Browseable, "Writeable =
yes" I saved it and reloaded Samba:

sudo systemctl restart smb.service

Part 2. Setting up Unreal Tournament 4 hub

I installed Unreal Tournament hub as a user, not as root. From what I understand, this is the only way to do this. If it is possible to run it as root, it could cause a security risk.
First, I had to download the UT4 Linux server package. I found the current version here:
https://s3.amazonaws.com/unrealtournament/ShippedBuilds/%2B%2BUT%2BRelease-Next-CL-3525360/UnrealTournament-Server-XAN-3525360-Linux.zip

The link is Linux (wget) friendly, but I downloaded onto my Windows 10 machine, and then transferred it to my Linux server via Samba to make sure that my Samba was working. I mapped the samba share to my Windows 10 machine by adding it as a network drive. It asked me for the location, which was \\192.168.1.99\My_User_Name\ and put in the credentials when asked. Alternatively, I could have logged into my console as MY_USER_NAME (not root) and downloaded it by typing:

wget https://s3.amazonaws.com/unrealtournament/ShippedBuilds/%2B%2BUT%2BRelease-Next-CL-3525360/UnrealTournament-Server-XAN-3525360-Linux.zip

Back on the Linux terminal, I unzipped the package:

unzip UnrealTournament-Server-XAN-3525360-Linux.zip

After unzipping, I did not delete the zip file. It may not be available in the future. I needed to make sure that I had read and write permissions to this new folder.

chown -R My_User_Name:My_User_Name *

I made a startup script to run the server in LinuxServer/Engine/Binaries/Linux/. I named it UT4Server.sh by typing:

cd LinuxServer/Engine/Binaries/Linux/
vi UT4Server.sh

Here is my copy. Alter it with notepad or equivalent to have your correct path to the server.

https://www.dropbox.com/scl/fi/b5wwfwt5zewvjl2hk128k/UT4Server.sh?rlkey=l4a0x56g4ycqfqywnageye59f&dl=0

I then ran 2 commands that would make all binaries within the new Server folder executable. First, I navigated to the Binaries folder:

cd /home/my_user_name/LinuxServer/Engine/Binaries/Linux/
and ran the command

chmod a+x *

Then I did the same for LinuxServer/UnrealTournament/Binaries/Linux/

cd /home/my_user_name/LinuxServer/Engine/Binaries/Linux/
chmod a+x *

I ran the server and it crashed, (as it was supposed to) while it builds some config file templates for us.

First I entered the directory where our startup script is stored:

cd /home/My_USER_NAME/LinuxServer/Engine/Binaries/Linux/

This was the command that I used:

./UT4Server.sh

When it crashed, it made a new folder and some files for me, as well as complaining that I had not read and accepted the EULA (End User License Agreement).
I pressed up, and pressed "Enter" and it ran the script again, this time with success.
I the opened my Unreal Tournament Game on my Windows 10 machine, and found it in the Hub list. I had to click the box that says "Show all empty Hubs/Servers" at the top right of the Hub Browser in my UT4 Game. I was able to join a very generic game. The name in the list was "UT Server" and the MOTD was "Have Fun". I played a quick round of Blitz and it ran fine.
Now it was time to setup my Game.ini so that it has a Name and MOTD in the hub. It was located at /LinuxServer/UnrealTournament/Saved/Config/LinuxServer I used Samba to go in and edit Game.ini to my liking. Here is a template that you can modify and use:

https://www.dropbox.com/scl/fi/vbhzp9ycirf92leyvxjle/Game.ini?rlkey=2optfep8eeppfkg6ng0aaaybg&dl=0

I also wanted to set a password so that I could login as an admin if needed, to kick, ban, or do other various functions. This is found in the /LinuxServer/UnrealTournament/Saved/Config/LinuxServer/Engine.ini:

[/Script/UnrealTournament.UTGameEngine] bFirstRun=False
RconPassword=MY_PASSWORD_HERE

Here is how to use it (in game):
While in game, open your console by pressing ~
After the console opens type in rconauth YOUR_PASSWORD
It will tell you that rcon is authenticated (meaning you are now logged in as admin)
Open your console by typing ~ again.
Type in rconexec adminmenu
From there you can do various commands like Kick or Ban.

The last thing that we must do in order to connect to the new Master server (timiimit server) is to append this to the end of your Engine.ini:

Vi /home/your_username_/LinuxServer/UnrealTournament/Saved/Config/LinuxServer/Engine.ini

Press i key then paste this to the bottom

[OnlineSubsystemMcp.BaseServiceMcp]
Domain=master-ut4.timiimit.com
Protocol=https
[OnlineSubsystemMcp.GameServiceMcp]
Domain=master-ut4.timiimit.com
Protocol=https
[OnlineSubsystemMcp.AccountServiceMcp]
Domain=master-ut4.timiimit.com
Protocol=https
[OnlineSubsystemMcp.OnlineFriendsMcp]
Domain=master-ut4.timiimit.com
Protocol=https
[OnlineSubsystemMcp.PersonaServiceMcp]
Domain=master-ut4.timiimit.com
Protocol=https
[OnlineSubsystemMcp.OnlineImageServiceMcp]
Domain=master-ut4.timiimit.com
Protocol=https

Then press escape key and type ":wq" and press enter.

Part 3. Setting up UTCC Redirect service

UTCC is the website where I will be gathering content from, and the players will automatically be getting that content from, when they connect to my Hub.

Here is a description of the site in their own words:
Welcome to UTCC!
Work in progress! UTCC is a platform created with the purpose of centralizing community efforts for Unreal Tournament. From sharing and finding content, to finding servers to play on or managing your own, UTCC strives to be the single resource for all of your needs." UTCC is purely a community effort and is not affiliated with Unreal Tournament or Epic Games. If you have an issue with anything on this site, please send us an email or contact @skandalouz#1109 on Discord."

First, we will have to register for an account:

Visit https://utcc.unrealpugs.com/register and use your email address, create a password and then verify the account through the email that is sent.
After that, login and get to work! https://utcc.unrealpugs.com/login takes us to the login page, login and then at the top right, click the ADD button.
On the page that opens, click the tab that says Server, and underneath put the name of your server. My UT4 Server is called "West Coast (Test Server)", so I put that there. I typed in a description, and clicked the "Add" button.
For the purpose of this tutorial, we are setting up a Deathmatch server. We won't be doing a full server setup, but just enough to see how the process works, test it and play it. After that, you can configure yours however you like.

Near the top left of the page is a clickable link called "Info". I clicked on that and found the link for the "Update Tool". I clicked that and found the download for the script. I followed the link and got the 2 files that I needed: config.yaml and updatescript.py

Edit. The Above paragraph is out of date, use this link instead (for now) until they update it on their site. You can use samba to retrieve this from the Dropbox link.
https://www.dropbox.com/scl/fi/epuzrlnogaufgu1os2vdm/UTCC.zip
I connected to the server via Samba, entered the LinuxServer, download the above file and unzipped it. unzip UTCC.zip
I then edited the configure.yaml file (instructions below) and placed both it, and the updatescript.py file inside it.

Config file for using the update script

THESE VALUES MUST BE ADDRESSED
private_code : "3X1K52332kbzCbbuM7Ugi3wGwj81oozU" <----- This is found under "my account" on UTCC website, under your name (top right), but its called "Private Token"
server_token : "O2g0M73B6BFKDF9922p9BdkoCCgIsSUD" <----- Find this by clicking your username again, clicking my content and then clicking your server.

hide_defaults : true
delete_old : true

FILE PATHS
server_loc : "/home/MY_USER_NAME/LinuxServer" <--- change this to match your /path/to/LinuxServer

After this, I figured it was a good time to back up my Game.ini in my /LinuxServer/UnrealTournament/
Using Samba, I copied and pasted the file, and named the backup Game.ini.bak

I needed to install the requests package that the UTCC site said I would need to run the python script:

sudo yum install python-yaml -y
sudo yum install gcc -y

The following commands had to be run as root.
Warning: Using Linux as the Root User
Caution should be exercised when operating as the root user in a Linux environment. The root user has elevated privileges and unrestricted access to system files, which can lead to unintended and potentially harmful consequences. It is advisable to refrain from regular usage as the root user and, instead, perform administrative tasks using the sudo command to minimize the risk of accidental system alterations. Always exercise caution and adhere to best practices to ensure the stability and security of your Linux system.There is only one set of commands in this tutorial that will require actual root login. After you issue these commands, be sure to log out afterwards.

su

Enter the root password.

scl enable rh-python36 bash
pip install requests
pip install pyyam
lpip install psutil

Control + D to log out of root.

We have to make a few empty files (if the script did not make them for you). I made mine in Windows, and used Samba to put them in the folders on the Linux server. If you must make them through terminal on the linux server you can use:

Mkdir

Folder_name

and vi
Text_document_name

to create files and folders.

Make an empty text document named "game_ini.txt" and put it in /LinuxServer/UTCC/Data/ if there isn't one there already.
Go into /LinuxServer/UnrealTournament/Saved/Config and make a folder called "Rulesets".
Place an empty text document named "rulesets.json" inside.

Next do chmod a+x * inside UTCC folder

Chmod a+x *

Before we run the script, we will kill the server if it is running:

ps ax

This will display a list of processes that are running. Find your PID (far left) and then do: kill 1234 (where 1234 is your PID) Now let's do:

scl enable rh-python36 bash

to switch back to the newest version of python.
Then we will run the script:

python updatescript.py

From the directory that it is in.
Double check your Game.ini to make sure that isn't erased, which was the issue that I had with the old script. If all went well, with no errors, we are ready to set up our update system.
Back to the UTCC site, I went to the top right of the page, and clicked my username. The list that opened had a click the "Add" button, and selected the "Ruleset" tab. I am running a Deathmatch server, so I named my Ruleset "Deathmatch". I then typed in a description and pressed "Add". Under "Ruleset Information" I edited it as follows:
Click the "Edit Button" at the top left.
Unique tag Classic DM
Categories Deathmatch

Title Classic Deathmatch
Tooltip Classic Deathmatch
Description 10 Player Deathmatch
Map Prefixes DM
Display Texture Deathmatch
The save it, top right “Save” button.

Under “Game Rules” I edited it in the same way:
Max Players 10
Gamemode
Deathmatch Then save.
Now let’s add a custom map.
At the lower part of the page, we have 3 sections. Type in what you want and click the result you are looking for:
Add map to Ruleset Cheops brought back DM-Cheops so I selected it. You can do the same thing for mutators, and game types, but for this tutorial we are keeping it simple.
We also need to connect the ruleset to the server by clicking your username, clicking my content, selecting your server, and then clicking on manage rulesets on the left, and then adding it to your server (you have to search for it by the name that you named it in the search box).
Now, we also need to go and find this custom map and add it to our server.
Top left of the website, I hit the browse button, clicked browse content, and then went down to the search field (bottom left), and typed in Cheops. It brought in DM-Cheops, so I clicked it.
It took me to the page for that map. I clicked “Connect to Server” on the left-hand side.
I selected my server and clicked on update.
I then went back to my server and ran the updatescript.py script:

./LinuxServer/UTCC/updatescript.py

Again, the download that their site has is outdated and doesn’t work. So, we are using a modified version that “mostly” works. I will update this tutorial when the new one is released.
The last workaround that we have to do, is open up our “Game.ini” and then copy the line from the redirect and paste it to a different part of your Game.ini
From:
[/Script/Engine.GameSession]
MaxPlayers=10
MaxSpectators=10
RedirectReferences=(PackageName="DM-Cheops-UT2004-WindowsNoEditor",PackageURLProtocol="http",PackageURL="utcc.unrealpugs.com/redirect/415/563e0
2bffcf072f0ec604f4dfca519d3/DM-Cheops-UT2004-WindowsNoEditor.pak",PackageChecksum="563e02bffcf072f0ec604f4dfca519d3")

To:
[/Script/UnrealTournament.UTBaseGameMode]
ServerInstanceID=MY_Server_Instance_ID_Goes_Here
RedirectReferences=(PackageName="DM-Cheops-UT2004-WindowsNoEditor",PackageURLProtocol="http",PackageURL="utcc.unrealpugs.com/redirect/415/563e0
2bffcf072f0ec604f4dfca519d3/DM-Cheops-UT2004-WindowsNoEditor.pak",PackageChecksum="563e02bffcf072f0ec604f4dfca519d3")
After doing this, I closed and saved the file, then started the server, connected to it with my UT4 game, and was able to start a game with Cheops, and it was automatically downloaded into my UT4 game. The very last thing I did was make a cron job so that anytime my server timed out, or my Centos 7 machine lost power and came back, unplanned or on a planned reboot, the UT4 server would automatically restart itself within a minute or so.

sudo yum install cronie -y

Then, as regular user:
crontab -e

That opens a file that you add this text to:
***/1 * * * * bash ./LinuxServer/Engine/Binaries/Linux/UT4Server.sh**

This is vi, so after pasting do:
"esc" then ":wq and enter".
That will write and save the file.

sudo + "**reboot**" or "**shutdown -r now**".

After the machine boots up, wait a few minutes and verify that your UT4 Server is up and running, and on the Hub / Server list on the game!

The end.

Notes to consider:

You can request that your hub become a "trusted" (players can gain XP points) by contacting a UT4MS admin on the UU discord - Permanent Invite Link:
https://discord.gg/2DaCWkK

The discord for UTCC is https://discord.gg/CMU93vhqyB and the admins there are very helpful, and as time goes on, you will want to be aware of any changes happening with that site. Especially if by the time you use this guide, something has changed or is no longer working.

Do not forget that just because you opened ports on the server in this tutorial, you still need to forward those ports to the server in your router or set the server as a DMZ (less secure).