

# Claude Code

- [Installation Claude Code](#)
- [Снятие лимита RDP-подключений в Windows Server 2025](#)
- [Глобальная настройка Claude Code на Windows Server](#)
- [Настройка алиаса для команды Claude](#)
- [Claude Code на linux-сервере](#)
  - [SSH-подключение](#)
  - [Синхронизация Syncthing](#)
- [Skills](#)
  - [Skill - Вставить всплывающий виджет GetCourse на лендинг](#)

# Installation Claude Code

1. Установить Node.js <https://nodejs.org/en/download>
2. Установить через терминал <https://code.claude.com/docs/en/overview>
3. Установить редактор кода, например VS <https://code.visualstudio.com/>

# ??????? RDP- ????????? Windows Server 2025

## ????????? ?????????? ?????????? Windows Server 2025 ??? ?????? (Grace Period 120 ????)

Эта инструкция описывает настройку одиночного сервера (Workgroup) для одновременной работы нескольких пользователей. Установка через "Quick Start" требует наличия домена (Active Directory), поэтому мы используем установку по ролям, чтобы обойти это ограничение.

---

## ??? 1: ?????????? ?????? Server Manager

Этот шаг превращает сервер в хост для удаленных сессий и устанавливает службу лицензирования для активации пробного периода.

1. Откройте **Server Manager**.
2. В правом верхнем углу нажмите **Manage** и выберите **Add Roles and Features**.
3. Нажмите **Next** на стартовом экране.
4. В разделе **Installation Type** выберите **Role-based or feature-based installation** (это критически важно для работы без домена) и нажмите **Next**.
5. В разделе **Server Selection** убедитесь, что выбран ваш сервер, и нажмите **Next**.
6. В разделе **Server Roles** найдите в списке **Remote Desktop Services** и поставьте галочку. Нажмите **Next**.
7. Пропустите раздел **Features**, просто нажав **Next** дважды, пока не дойдете до **Role Services**.
8. В разделе **Role Services** отметьте две службы:
  - **Remote Desktop Session Host**
  - **Remote Desktop Licensing**

- (Если появится всплывающее окно с предложением добавить компоненты — нажимайте **Add Features**).

9. Нажмите **Next**, затем **Install**.
10. Дождитесь окончания установки и **обязательно перезагрузите сервер**.

---

## ??? 2: ?????????? ??????? Group Policy Editor

После перезагрузки сервер нужно перевести в режим 120-дневного пробного периода и снять лимит на подключения. Все настройки выполняются в английском интерфейсе.

1. Нажмите `Win + R`, введите `gpedit.msc` и нажмите **Enter**.
2. В левом меню перейдите по пути:

`Computer Configuration` -> `Administrative Templates` -> `Windows Components` -> `Remote Desktop Services` -> `Remote Desktop Session Host`.

### Настройка лимита сессий:

1. Откройте папку **Connections**.
2. Дважды кликните на **Limit number of connections**. Установите значение **Enabled**, а в поле "RD Maximum Connections allowed" введите `999999` (или любое нужное число). Нажмите **OK**.
3. (Если нужно, что бы под одним юзером были несколько сессий) Дважды кликните на **Restrict Remote Desktop Services users to a single Remote Desktop Services session**. Установите **Disabled** и нажмите **OK**.

### Настройка сервера лицензирования (Grace Period):

1. В левом меню перейдите в папку **Licensing** (находится рядом с Connections).
2. Откройте **Use the specified Remote Desktop license servers**. Установите **Enabled** и в поле "License servers to use" впишите `127.0.0.1`. Нажмите **OK**.
3. Откройте **Set the Remote Desktop licensing mode**. Установите **Enabled** и в выпадающем списке выберите **Per User**. Нажмите **OK**.

---

## ??? 3: ?????????? ?????????????????????? ? ?????????? ?????

Для обеспечения отдельной авторизации и изоляции рабочих сред каждого сотрудника необходимо создать локальные учетные записи.

1. Нажмите `Win + X` и выберите **Computer Management**.
  2. В левом меню разверните **Local Users and Groups** и выберите папку **Users**.
  3. Нажмите правой кнопкой мыши в пустом месте по центру и выберите **New User**.
  4. Заполните поля **User name** и **Password** для каждого сотрудника. Снимите галочку "User must change password at next logon" и поставьте "Password never expires" (по желанию). Создайте нужное количество пользователей.
  5. Перейдите в папку **Groups** (слева).
  6. Найдите в списке группу **Remote Desktop Users** и кликните по ней дважды.
  7. Нажмите кнопку **Add**, введите имена созданных пользователей (можно через точку с запятой) и нажмите **Check Names**, затем **OK**.
- 

## ??? 4: ?????????? ?????????? ? ??????????

1. Чтобы заставить сервер сразу применить новые политики без перезагрузки, нажмите `Win + X` -> **Terminal (Admin)** или **Command Prompt (Admin)**.
  2. Введите команду обновления политик:  
`gpupdate /force`
  3. Нажмите Enter и дождитесь сообщения об успешном выполнении.
  4. Откройте поиск Windows, введите **RD Licensing Diagnoser** и запустите утилиту. В окне должно быть указано, что сервер лицензирования доступен, а текущий режим — **Grace Period** (120 дней).
-



Этот `.bat` файл самостоятельно найдет (или установит) Git, создаст нужные папки, выдаст права и пропишет системные пути. Кодировка исправлена для корректного отображения кириллицы.

1. Создайте на рабочем столе файл `setup_claude_global.bat`.
2. Скопируйте в него код ниже.
3. Нажмите на файл правой кнопкой мыши и выберите **«Запуск от имени администратора»**.

Фрагмент кода

```
@echo off
:: Set encoding for correct display of English text
chcp 866 >nul
setlocal enabledelayedexpansion
title Claude Code Global Setup

echo =====
echo  Claude Code: Global Setup (Multi-User)
echo =====

:: 1. Check for administrator rights
net session >nul 2>&1
if %errorLevel% neq 0 (
    echo [ERROR] Run this file AS ADMINISTRATOR.
    pause
    exit /b
)

:: 2. Search for or install Git Bash (FIXED LOGIC)
set "BASH_PATH="

if exist "C:\Program Files\Git\bin\bash.exe" (
    set "BASH_PATH=C:\Program Files\Git\bin\bash.exe"
)

if "!BASH_PATH!"==" (
    if exist "C:\Program Files (x86)\Git\bin\bash.exe" (
        set "BASH_PATH=C:\Program Files (x86)\Git\bin\bash.exe"
    )
)
```

```
if "!BASH_PATH!"==" (
    echo [STATUS] Git Bash not found. Starting installation via winget...
    winget install --id Git.Git -e --source winget --accept-source-agreements --accept-
package-agreements
    set "BASH_PATH=C:\Program Files\Git\bin\bash.exe"
)

echo [STATUS] Git Bash path set to: !BASH_PATH!

:: 3. Configure shared folders
set "NPM_GLOBAL=C:\ProgramData\npm"
set "CLAUDE_CONFIG=C:\ProgramData\claude-code"

if not exist "%NPM_GLOBAL%" mkdir "%NPM_GLOBAL%"
if not exist "%CLAUDE_CONFIG%" mkdir "%CLAUDE_CONFIG%"

:: 4. Configure access permissions (icacls)
echo [STATUS] Setting access permissions for the Users group...
icacls "%NPM_GLOBAL%" /grant Users:(OI)(CI)M /T /C /Q >nul
icacls "%CLAUDE_CONFIG%" /grant Users:(OI)(CI)M /T /C /Q >nul

:: 5. System variables (Machine level)
setx CLAUDE_CONFIG_DIR "%CLAUDE_CONFIG%" /M >nul
setx CLAUDE_CODE_GIT_BASH_PATH "!BASH_PATH!" /M >nul

:: 6. Move NPM and install Claude
call npm config set prefix "%NPM_GLOBAL%" --global >nul
echo [STATUS] Installing Claude Code (please wait)...
call npm install -g @anthropic-ai/claude-code >nul

:: 7. Update system PATH
set "NEW_PATH=%NPM_GLOBAL%"
echo %PATH% | findstr /I /C:"%NEW_PATH%" >nul
if %errorlevel% neq 0 (
    setx /M PATH "%PATH%;%NEW_PATH%" >nul
)

echo.
echo =====
```

```
echo  INSTALLATION COMPLETED SUCCESSFULLY!
echo =====
echo Restart the terminal and enter 'claude auth'.
pause
```

## ??????? 2: ??????? ????????????

# (PowerShell)

Если вы хотите выполнить все шаги вручную, откройте **PowerShell от имени Администратора**:

??? 1: ?????????????? ?????? ? ??????

PowerShell

```
$dirs = @"C:\ProgramData\npm", "C:\ProgramData\claude-code"
foreach ($dir in $dirs) {
    if (!(Test-Path $dir)) { New-Item -ItemType Directory -Path $dir }
    $acl = Get-Acl $dir
    $rule = New-Object System.Security.AccessControl.FileSystemAccessRule("Users", "Modify",
"ContainerInherit, ObjectInherit", "None", "Allow")
    $acl.AddAccessRule($rule)
    Set-Acl $dir $acl
}
```

??? 2: ?????????????? ?????????????? ????????????????

PowerShell

```
[Environment]::SetEnvironmentVariable("CLAUDE_CONFIG_DIR", "C:\ProgramData\claude-code",
"Machine")
[Environment]::SetEnvironmentVariable("CLAUDE_CODE_GIT_BASH_PATH", "C:\Program
Files\Git\bin\bash.exe", "Machine")

$oldPath = [Environment]::GetEnvironmentVariable("Path", "Machine")
```

```
if ($oldPath -notlike "*C:\ProgramData\npm*") {  
    [Environment]::SetEnvironmentVariable("Path", "$oldPath;C:\ProgramData\npm", "Machine")  
}
```

??? 3: ?????????? ??????????

PowerShell

```
npm config set prefix "C:\ProgramData\npm" --global  
npm install -g @anthropic-ai/claude-code
```

????????????? ????: ??????????????  
?????????????????

Независимо от выбранного метода установки, после его завершения:

1. Откройте **новое** окно терминала.
2. Введите `claude auth`.
3. Пройдите процесс авторизации в браузере. Файл сессии сохранится в общую папку.

????????????? ??? ?????????????????????? (   
?????????????, David)

Для любого пользователя сервера процесс теперь максимально прост:

1. Пользователь заходит в свою учетную запись.
2. Открывает терминал (PowerShell, CMD или терминал в IDE).
3. Вводит `claude`.
4. Программа запускается **сразу под вашим аккаунтом**.

# ????????????? ?? ?????????? ??????????

## (Troubleshooting)

Симптом	Причина	Решение
<b>Ошибка:</b> <code>Command 'claude' not found</code> у нового пользователя.	Windows еще не обновила системный <code>PATH</code> для текущей сессии пользователя.	Пользователю нужно выйти из системы (Log off) и зайти снова, либо перезагрузить сервер.
<b>Ошибка:</b> <code>EPERM: operation not permitted</code> при запуске <code>claude</code> .	Программа все еще пытается получить доступ к папке <code>AppData</code> администратора.	Убедитесь, что шаг с <code>npm config set prefix "C:\ProgramData\npm" --global</code> был выполнен, и переустановите Claude.
<b>Ошибка:</b> <code>Unable to find CLAUDE_CODE_GIT_BASH_PATH</code> .	Неверный путь к <code>bash.exe</code> или Git не установлен.	Проверьте фактическое расположение файла <code>bash.exe</code> и обновите системную переменную <code>CLAUDE_CODE_GIT_BASH_PATH</code> .
<b>Требуется</b> <code>claude auth</code> у второго пользователя.	Переменная <code>CLAUDE_CONFIG_DIR</code> установлена как "User", а не "Machine".	Запустите PowerShell от администратора и выполните команду из Шага 2 ручной настройки.

# ????????????? ?????????????? Claude Code

Поскольку программа теперь находится в общей системной папке, **обновлять ее должен только Администратор**.

В терминале от имени Администратора введите:

PowerShell

```
npm install -g @anthropic-ai/claude-code@latest
```

Ваши ключи авторизации не пострадают, а новые файлы автоматически унаследуют права доступа для всех пользователей.

# ?????????? ??????? ???? ????????? Claude

Чтобы сделать алиас **claude-skip** на Windows для всех пользователей, выполните эти **2 шага** в PowerShell от имени Администратора:

## 1. ?????????? ??????????

Запустите этот код, чтобы создать системный батник (он будет работать везде: в CMD, PowerShell и IDE):

PowerShell

```
"@echo off`nclaude --dangerously-skip-permissions %*" | Out-File -FilePath  
"$env:SystemRoot\claude-skip.bat" -Encoding ASCII
```

## 2. ?????????? ??????????????

Откройте любое новое окно терминала и введите:

PowerShell

```
claude-skip --version
```

---

**Готово!** Теперь вместо длинной команды просто пишете `claude-skip`.

Claude Code ?? linux-????????

# SSH-??????????

??????????

Host: **62.84.181.155** ??? **22.claude.fitnessmama.md**

???? ?? ??????: **ssh <username>@<host>**

**Пример:**

```
ssh olesya@62.84.181.155
```

?????????? ssh-??????

**Примечание:** в примерах имя ключа - `claude_key`. Если вы назовёте ключ иначе, не забудьте учесть это в тех командах, где встречается "`claude_key`" или "`claude_key.pub`"

## Windows

- Запускаем Powershell **от имени администратора**.
- Создаём папку для ключей:

```
if (!(Test-Path $env:USERPROFILE\.ssh)) {  
    Write-Host "Папка .ssh не найдена – создаём её..." -ForegroundColor Yellow  
    New-Item -ItemType Directory -Path $env:USERPROFILE\.ssh  
    Write-Host "Папка .ssh создана" -ForegroundColor Green  
} else {  
    Write-Host "Папка .ssh уже существует" -ForegroundColor Green  
}
```

- Создаём SSH-ключ:

```

$keyPath = "$env:USERPROFILE\.ssh\claude_key"
if (!(Test-Path $keyPath)) {
    Write-Host "Создаём новый SSH-ключ..." -ForegroundColor Yellow
    ssh-keygen -t ed25519 -f $keyPath -N ""
    Write-Host "SSH-ключ создан!" -ForegroundColor Green
} else {
    Write-Host "SSH-ключ уже существует: $keyPath" -ForegroundColor Green
}

```

? ??????????? C:\Users\<??? ?????????>\.ssh ?????????? ?????? claude\_key ?  
 claude\_key.pub

## MacOS/Linux

- **Открываем терминал.**
- **Создаем папку для ключей:**

```

SSH_DIR="$HOME/.ssh"

if [ ! -d "$SSH_DIR" ]; then
    echo "Папка .ssh не найдена – создаём её..."
    mkdir -p "$SSH_DIR"
    chmod 700 "$SSH_DIR"
    echo "Папка .ssh создана"
else
    echo "Папка .ssh уже существует"
fi

```

- **Создаем SSH-ключ:**

```

KEY_PATH="$HOME/.ssh/claude_key"

if [ ! -f "$KEY_PATH" ]; then
    echo "Создаём новый SSH-ключ..."
    ssh-keygen -t ed25519 -f "$KEY_PATH"
    echo "SSH-ключ создан!"
else

```

```
echo "SSH-ключ уже существует: $KEY_PATH"
fi
```

```
? ?????????? /home/<??? ??????????>/.ssh ?????????? ?????? claude_key ?
claude_key.pub
```

???????????????? ????????????????? ??????? ?? ?????????

## Windows

**У себя на компьютере в Powershell:**

```
type $env:USERPROFILE\.ssh\claude_key.pub | ssh <user>@62.84.181.155 "cat >>
~/.ssh/authorized_keys"
```

## MacOS/Linux

**В терминале:**

```
cat ~/.ssh/claude_key.pub | ssh <user>@62.84.181.155 "cat >> ~/.ssh/authorized_keys"
```

???????????????? ????????????????? ?????????

### Windows

**В Powershell от имени администратора:**

```
# Путь к SSH-директории и config
$sshDir = "$env:USERPROFILE\.ssh"
$configPath = "$sshDir\config"
```

```
if (!(Test-Path $sshDir)) {
    New-Item -ItemType Directory -Path $sshDir | Out-Null
}

if (!(Test-Path $configPath)) {
    New-Item -ItemType File -Path $configPath | Out-Null
}

# Блок для добавления (ЗАМЕНИ <username>)
$block = @"
Host claude
    HostName 22.claude.fitnessmama.md
    User <username>
    IdentityFile ~/.ssh/claude_key
    IdentitiesOnly yes
"@

$content = Get-Content $configPath -Raw
if ($content -notmatch "Host claude") {
    Add-Content -Path $configPath -Value "`n$block"
    Write-Host "Конфигурация добавлена"
} else {
    Write-Host "Host claude уже есть в config"
}

icacls $configPath /inheritance:r | Out-Null
icacls $configPath /grant:r "$($env:USERNAME):(R,W)" | Out-Null
```

## MacOS/Linux

### В терминале:

```
# Путь
SSH_DIR="$HOME/.ssh"
CONFIG_FILE="$SSH_DIR/config"
```

```
mkdir -p "$SSH_DIR"
chmod 700 "$SSH_DIR"

touch "$CONFIG_FILE"
chmod 600 "$CONFIG_FILE"

# Добавляем блок (ЗАМЕНИ <username>)
if ! grep -q "Host claude" "$CONFIG_FILE"; then
cat >> "$CONFIG_FILE" <<EOF

Host claude
    HostName 22.claude.fitnessmama.md
    User <username>
    IdentityFile ~/.ssh/claude_key
    IdentitiesOnly yes
EOF
    echo "Конфигурация добавлена"
else
    echo "Host claude уже есть в config"
fi
```

**После создания такого конфига можно не писать `ssh <username>@<host>` или `scp <username>@<host>`.**

**Достаточно `ssh claude` или `scp claude`. Пароль так же не будет нужен, если не выполняете команды на сервере с `sudo`**

Claude Code на linux-сервере

????????????? Syncning

# Skills

# Skill - ?????????? ?????????????????? ????????? GetCourse ?? ??????????

---

description: Вставить всплывающий виджет GetCourse на лендинг. Передай скрипт(ы) виджета и укажи, какая кнопка должна его открывать.

---

## ## Что такое виджет GetCourse

Виджет GetCourse – это тег `<script>` с двумя ключевыми атрибутами:

- `id` – уникальный хеш, по которому платформа идентифицирует точку вставки
- `src` – URL вида `https://fitnessmama.school/pl/lite/widget/script?id=XXXXXXX`

Скрипт рендерит форму регистрации/оплаты прямо рядом с собой в DOM.

## ## Твоя задача

Пользователь предоставил один или несколько скриптов виджетов GetCourse и указал, какие кнопки должны их открывать.

Выполни следующие шаги:

### ### 1. Прочитай файл лендинга

Прочитай целевой HTML-файл, чтобы понять:

- Где находится закрывающий тег `</style>`
- Какие кнопки нужно связать с виджетами (их текст, классы, текущий `href`)
- Есть ли уже существующие модальные окна или скрипты виджетов

### ### 2. Добавь CSS модальных окон

Вставь перед `</style>` следующие стили (адаптируй `border-radius` и цвета под дизайн страницы):

```
```css
```

```
/* ===== GETCOURSE WIDGET MODALS ===== */
.gc-modal-overlay {
  display: none;
  position: fixed;
  inset: 0;
  background: rgba(0,0,0,0.6);
  backdrop-filter: blur(4px);
  z-index: 9999;
  align-items: center;
  justify-content: center;
}
.gc-modal-overlay.active {
  display: flex;
}
.gc-modal {
  background: #fff;
  border-radius: 16px;
  width: 100%;
  max-width: 500px;
  margin: 16px;
  overflow: hidden;
  position: relative;
}
.gc-modal-close {
  position: absolute;
  top: 12px;
  right: 16px;
  font-size: 28px;
  cursor: pointer;
  color: #666;
  background: none;
  border: none;
  line-height: 1;
  z-index: 1;
}
.gc-modal-close:hover {
  color: #222;
}
.gc-modal-body {
  padding: 8px;
```

```
}  
...  

```

### ### 3. Добавь HTML модальных окон и JavaScript

Вставь перед `</body>`. Для каждого виджета создай отдельный оверлей.  
Именуй id как `gc-modal-[тип]`, например `gc-modal-free`, `gc-modal-vip`.  
Скрипт виджета помести внутрь `.gc-modal-body`.

Шаблон для одного виджета:

```
```html  
<!-- Modal [тип] -->  
<div class="gc-modal-overlay" id="gc-modal-[тип]">  
  <div class="gc-modal">  
    <button class="gc-modal-close" onclick="gcCloseModal('[тип]')">&times;</button>  
    <div class="gc-modal-body">  
      <script id="[ID_ВИДЖЕТА]" src="[SRC_ВИДЖЕТА]"></script>  
    </div>  
  </div>  
</div>  
...  

```

После всех модалок добавь единый блок JavaScript:

```
```html  
<script>  
  function gcOpenModal(type) {  
    document.getElementById('gc-modal-' + type).classList.add('active');  
    document.body.style.overflow = 'hidden';  
  }  
  function gcCloseModal(type) {  
    document.getElementById('gc-modal-' + type).classList.remove('active');  
    document.body.style.overflow = '';  
  }  
  // Закрытие по клику на фон оверлея  
  document.querySelectorAll('.gc-modal-overlay').forEach(function(overlay) {  
    overlay.addEventListener('click', function(e) {  
      if (e.target === overlay) {  
        overlay.classList.remove('active');  
      }  
    });  
  });  
</script>  

```

```
        document.body.style.overflow = '';
    }
});
});
// Закрытие по клавише Escape
document.addEventListener('keydown', function(e) {
    if (e.key === 'Escape') {
        document.querySelectorAll('.gc-modal-overlay.active').forEach(function(m) {
            m.classList.remove('active');
        });
        document.body.style.overflow = '';
    }
});
</script>
...

```

### ### 4. Измени кнопки

Для каждой кнопки, которую нужно связать с виджетом:

- Замени `href="#"` на `href="javascript:void(0)"`
- Добавь `onclick="gcOpenModal('[тип]')`

Пример:

```
```html
<!-- До -->
<a href="#" class="btn-primary">Купить</a>

<!-- После -->
<a href="javascript:void(0)" class="btn-primary" onclick="gcOpenModal('vip')">Купить</a>
...

```

### ### 5. Проверь результат

Сообщи пользователю:

- Какие кнопки были изменены
- Какие модальные окна созданы (id каждого)
- Как протестировать: открыть файл в браузере, нажать кнопки, проверить закрытие через x, клик на фон, Escape

## Важные правила

- **Не дублируй** скрипты виджетов – каждый `id` виджета должен встречаться в DOM только один раз
- Если на странице уже есть своя модальная система, **используй её стили** вместо gc-modal-\*
- Если пользователь указал несколько виджетов (Free, VIP и т.д.) – создай отдельный оверлей для каждого
- Если кнопка уже ссылается на `#tariffs` или другой якорь (прокрутка к секции) – **не трогай её**, только кнопки самой оплаты/регистрации