# Absolute Raid Bot: Setting Up Azure

Hosting guide for Absolute Raid Bot 1.0.6: run your guild's bot 24/7 on a small Linux virtual machine in Azure.

Absolute Gaming LLC. Written by ApeTheory - Admin for Absolute Gaming LLC.

## Before you start: Azure costs money

> **Running a virtual machine in Azure is not free.** Azure is a paid service from Microsoft. You need a Microsoft account with a **credit or debit card on file** (prepaid cards are not accepted), and Microsoft bills that card every month for what the VM uses. Absolute Gaming LLC does not pay for, see, or manage your Azure account or bill.

What to expect, at Microsoft's pay-as-you-go prices in a US region (check the current numbers in the Azure Pricing Calculator before you build):

| Item | What it is | Roughly per month |
|---|---|---|
| Standard_B1s VM | 1 vCPU, 1 GB RAM, Linux | about $7.60 |
| OS disk | 30 GB Standard SSD | a few dollars |
| Public IP address | Standard, static | about $3.65 |
| Network traffic | the bot uses very little | usually pennies |
| **Total** | | **about $12 to $15** |

A few things that change the bill:

- **New accounts:** Microsoft's free account gives you a $200 credit for the first 30 days. Its free-services list also includes 750 hours a month of B1s VMs, enough for one VM running all month. To keep using it after 30 days (or when the credit runs out) you must move to pay-as-you-go. Microsoft's offer terms change, so read the current ones at azure.microsoft.com/free.
- **Stopping vs deallocating:** shutting the VM down from inside Linux still bills compute. Only **Stop** in the Azure portal (status "Stopped (deallocated)") stops compute charges. The disk and public IP still bill while it's stopped.
- **Deleting:** delete the whole resource group when you are done, or charges continue.
- **Set a budget alert** (step 2.5) so a surprise never reaches your card.

## 1. What you are building

```
Discord (discord.com, gateway.discord.gg)
       ^  outbound HTTPS / secure WebSocket (TCP 443), started by the bot
       |
[ Azure network security group ]  in: SSH from your IP only   out: 443 and 80 only
       |
[ Ubuntu 24.04 VM, Standard_B1s ]
    absolute-raid-bot.service  (systemd: starts on boot, restarts on crash)
    ufw host firewall: deny all incoming except SSH
    /opt/absolute-raid-bot/data   database, audit log, backups, secret.key
```

The bot **only connects out.** Discord never connects in to it, so no port is opened for Discord. The only way in is SSH, and only from your own IP address.

On first boot the VM makes a one-time **pairing code.** Until someone with Manage Server enters that code in your Discord server, the bot answers nothing and works nowhere. After pairing it works only in that one server and leaves any other server it is added to.

## 2. Create an Azure account

Skip this section if you already have an Azure subscription you can use.

### 2.1 What you need

- A Microsoft account (Outlook, Hotmail, Xbox and similar logins all work) or a GitHub account. You can make a new one during sign-up.
- A phone number that can get a text or call, for identity checks.
- A credit card or non-prepaid debit card. Microsoft may place a temporary $1 verification hold on it, which is reversed.

### 2.2 Sign up

1. Go to **azure.microsoft.com/free** and choose **Try Azure for free** (or **Pay as you go** if you already used your free account).
2. Sign in with your Microsoft or GitHub account, or create one.
3. **About you:** enter your country, name, email and phone. Pick the country you live in; it can't be changed later.
4. **Identity verification by phone:** enter the code Microsoft texts or calls you with.
5. **Identity verification by card:** enter the card. The free account says it won't charge you until you move to pay-as-you-go, but the card must be valid.
6. Accept the agreement and select **Sign up**. Azure opens the portal at **portal.azure.com**.

### 2.3 Check you have a subscription

In the portal search bar type **Subscriptions** and open it. You should see one subscription (for example "Azure subscription 1") with status **Active**. Everything in this guide is created inside it.

### 2.4 Turn on multi-factor authentication

Your Azure login can create things that cost money, so protect it. Sign in at **account.microsoft.com/security**, then turn on two-step verification with the Microsoft Authenticator app.

### 2.5 Set a budget alert (recommended)

1. Search **Cost Management** and open **Cost Management + Billing**, then **Cost Management**, then **Budgets**.
2. Select **Add**. Scope: your subscription. Name: `raidbot-budget`. Reset period: Monthly. Amount: for example `20`.
3. Add alert conditions, for example 80% and 100% of Actual cost, and your email address. Select **Create**.

Azure emails you when spending crosses those lines. A budget alert does not stop the VM by itself; it warns you.

## 3. Get your Discord details ready

You will type these into the VM in step 6. **Setting Up the Discord Bot** (sections 3 and 4) shows each step in Discord.

- **Bot token** from the Discord Developer Portal (Bot page, Reset Token). Treat it like a password. Never paste it into Discord, a ticket, or a screenshot.
- **Server ID:** turn on Developer Mode in Discord (User Settings, Advanced), then right-click your server and choose **Copy Server ID**.
- **Owner user ID** (optional): right-click yourself and choose **Copy User ID**. Leave it empty and whoever pairs the bot becomes its owner.
- **Staff log channel ID** (optional): right-click the channel and choose **Copy Channel ID**.
- The bot must already be **invited to your server** with the `bot` and `applications.commands` scopes.

## 4. What you download, and how you build the VM

**There is no VM file to download and upload.** An Azure VM disk is a 30 GB file that has to be uploaded in Azure's own format, and an Azure image can only be used inside the subscription that made it. Instead you create a small, ordinary Ubuntu VM in the Azure portal, and it **sets itself up**: on its first boot it downloads Absolute Raid Bot from absolutegaming.store, checks the download's fingerprint, installs it, locks down its firewall and makes its pairing code.

What comes from the website:

| Download | What it's for |
|---|---|
| **Azure quick setup** (`absolute-raid-bot-cloud-init.yaml`) | Paste it into the VM's **Custom data** box when you create the VM. The VM installs the bot by itself. No file uploads, no install commands |
| **Absolute Raid Bot** (`AbsoluteRaidBot-1.0.6.zip`) | The bot, for a manual install or your own PC |
| **Azure VM template** (`AbsoluteRaidBot-Azure-Template-1.0.6.zip`) | The quick-setup file, the install scripts, a Packer image template and a PowerShell deploy script |

Choose a path:

| Path | Who it's for | What you do |
|---|---|---|
| **A. Quick setup** (recommended) | Any guild officer | Create the VM in the portal (section 5A) and paste the quick-setup file into **Custom data**. Wait about 10 minutes, then run `sudo arb-setup` |
| **B. Manual install** | If you skipped Custom data, or want to see each step | Create the VM the same way, upload the bot zip and run one install command (section 5A.5) |
| **C. Pre-built image** | Someone who runs bots for many guilds in one Azure subscription | Build the image once with Packer and deploy VMs from it with the PowerShell script (section 5B) |

All three end with the same VM: the bot installed as a service that starts on boot and restarts if it stops, the firewall locked down, and a pairing code waiting.

**What's preconfigured, and what you add.** The VM comes ready to run: Python, the bot, its service, automatic security updates, the firewall rules and the admin tools. You add only your guild's details, by running `sudo arb-setup` once: the Discord bot token, your server ID and, optionally, the owner and a log channel. The token is never in the website downloads or in the quick-setup file, because anyone holding it could control your bot.

# 5A. Paths A and B: create the VM in the Azure portal

## 5A.1 Create the virtual machine

1. In **portal.azure.com** search **Virtual machines**, then select **Create**, then **Azure virtual machine**.
2. **Basics** tab:
   - Subscription: yours. Resource group: **Create new**, `rg-raidbot`.
   - Virtual machine name: `vm-raidbot`.
   - Region: one near you with B1s available, for example **(US) West US 3** or **(US) East US**.
   - Availability options: **No infrastructure redundancy required**.
   - Security type: leave the default (Trusted launch).
   - Image: **Ubuntu Server 24.04 LTS - x64 Gen2**.
   - Size: select **See all sizes**, search `B1s`, choose **Standard_B1s** (1 vCPU, 1 GiB).
   - Authentication type: **SSH public key**. Username: `arbadmin`. SSH public key source: **Generate new key pair**. Key pair name: `vm-raidbot_key`.
   - Public inbound ports: **Allow selected ports**, then **SSH (22)** only. You lock it to your IP in step 5A.3.
3. **Disks** tab: OS disk type **Standard SSD (locally-redundant storage)**. Keep **Delete with VM** checked.
4. **Networking** tab:
   - Leave the new virtual network and subnet Azure suggests.
   - Public IP: keep the new one. **The bot needs it** to reach Discord: virtual networks created since March 31, 2026 are private by default, and the public IP is the VM's way out.
   - NIC network security group: **Basic**, public inbound ports **SSH (22)**.
   - Check **Delete public IP and NIC when VM is deleted.**
5. **Management** tab:

- **Auto-shutdown: leave it OFF.** It would turn the bot off every night.
- Leave automatic guest OS patching at its default; the VM also installs security updates itself.

6. **Monitoring** tab: Boot diagnostics **Enable with managed storage account** (the default). It lets you see the VM's console, and where you can read the pairing code without signing in.
7. **Advanced** tab (**Path A, quick setup**): open **absolute-raid-bot-cloud-init.yaml** from the website in Notepad, press **Ctrl+A**, **Ctrl+C**, and paste it into **Custom data**. Leave everything else as it is. (Path B: leave Custom data empty.)
8. **Review + create**, check the price shown, then **Create**.
9. When asked, select **Download private key and create resource**. Save `vm-raidbot_key.pem` somewhere safe (for example `Documents\Azure`). It is the only key to the VM and Azure can't show it again.
10. When deployment finishes select **Go to resource** and copy the **Public IP address**.

## 5A.2 Connect from your PC

Windows 10 and 11 include the SSH client. Open **PowerShell**:

```
cd $HOME\Documents\Azure
icacls .\vm-raidbot_key.pem /inheritance:r /grant:r "$($env:USERNAME):(R)"
ssh -i .\vm-raidbot_key.pem arbadmin@<public IP>
```

The `icacls` line makes the key readable only by you; SSH refuses keys that other accounts can read. Type `yes` the first time SSH asks about the host fingerprint. On macOS or Linux use `chmod 600 vm-raidbot_key.pem` instead of `icacls`.

## 5A.3 Lock SSH to your IP address

1. In the portal open the VM, then **Networking** (Network settings).
2. Open the inbound rule for port 22 (named like `SSH`).
3. Source: **My IP address** (the portal fills in your current IP). Save.

If your home IP changes, update that rule again (or you'll get a timeout when connecting).

## 5A.4 Add the outbound firewall rules

Same Networking page, **Outbound port rules**, **Add** each of these (Destination: **Service Tag**, **Internet**):

| Priority | Name | Port | Protocol | Action |
|---|---|---|---|---|
| 100 | Allow-HTTPS-Out | 443 | TCP | Allow |
| 110 | Allow-HTTP-Out-Updates | 80 | TCP | Allow |
| 4000 | Deny-Other-Internet-Out | * | Any | Deny |

Section 7 explains every rule.

## 5A.5 Install the bot

**Path A (quick setup):** nothing to do. The VM installs the bot on its first boot, which takes about 10 minutes after the deployment finishes. To see when it's done:

- In the portal: the VM -> **Help** -> **Boot diagnostics** -> **Serial log**. Near the end you'll see a box with **Absolute Raid Bot is installed. PAIRING CODE: ...** (select **Refresh** if it isn't there yet).
- Or over SSH (step 5A.2): run `cloud-init status --wait`. It returns `status: done` when the install has finished, and the login banner then shows the bot's state.

If something went wrong, `sudo cat /var/log/cloud-init-output.log` on the VM shows every step. Go to **section 6**.

**Path B (manual install):** on your PC, in a **new** PowerShell window (in the folder with the zip and the key):

```
scp -i .\vm-raidbot_key.pem .\AbsoluteRaidBot-1.0.6.zip arbadmin@<public IP>:~
```

Then in your SSH window on the VM:

```
sudo apt-get update && sudo apt-get install -y unzip
unzip AbsoluteRaidBot-1.0.6.zip -d raidbot
cd raidbot
sudo bash azure/scripts/install.sh "$PWD/absolute-raid-bot"
```

The installer takes a few minutes. It:

- installs Python, creates the `arb` service account, and copies the bot to `/opt/absolute-raid-bot`
- builds the Python environment and runs the bot's self-tests (the install stops if any fail)
- installs the service plus the tools `arb-setup`, `arb-status`, `arb-reset-2fa` and `arb-pair-reset`
- makes the **pairing code**
- turns on automatic security updates
- turns on the ufw firewall: deny all incoming except SSH

It ends with `Absolute Raid Bot installed in /opt/absolute-raid-bot. Run: sudo arb-setup`. You can delete the `raidbot` folder and the zip afterwards. Go to **section 6**.

## 5B. Path C: build the image and deploy from it

For whoever runs many bot VMs in one subscription. Needs the Azure CLI (`az`), Packer 1.9 or newer, and PowerShell 7 on your PC.

### 5B.1 Build the image with Packer

The template is on the website as **AbsoluteRaidBot-Azure-Template-1.0.6.zip** (and in the bot zip, in `azure/`). Either way, `absolute-raid-bot/` and `azure/` must sit side by side (unzip the template into the folder where you unzipped the bot). Run Packer from `azure/packer`.

```
az login
az group create -n rg-absolute-raid-bot-images -l westus3
cd azure/packer
copy raidbot.pkrvars.hcl.example raidbot.pkrvars.hcl     (then put your subscription ID in it)
packer init .
packer build -var-file=raidbot.pkrvars.hcl .
```

What the template (`absolute-raid-bot.pkr.hcl`) does:

- starts a temporary Ubuntu 24.04 LTS VM from Canonical's `ubuntu-24_04-lts` image (SKU `server`), signing in with your `az login`
- uploads the bot and the scripts, and runs `install.sh` with `ARB_IMAGE_BUILD=1`, so **no pairing code is made inside the image**
- generalizes it (`waagent -deprovision+user`), so every VM made from it gets its own identity
- saves a managed image named like `absolute-raid-bot-20261110-1500` in your resource group, and deletes the temporary VM

The build VM is a B2s for speed and only runs for the build (about 10 to 15 minutes). The image itself has no token, no database and no pairing code.

### 5B.2 Deploy a VM from the image

```
$img = az image list -g rg-absolute-raid-bot-images --query "[-1].id" -o tsv
./azure/deploy/New-RaidBotVM.ps1 -ResourceGroup rg-raidbot-<guild> -ImageId $img -Location westus3
```

The script (`New-RaidBotVM.ps1`) creates, in order:

- the resource group, a virtual network `10.50.0.0/24` with subnet `bot` (`10.50.0.0/27`)
- the network security group with the rules in section 7, attached to the subnet (SSH is allowed only from your current public IP, which it looks up, or `-AdminIp` if you give one)
- a static Standard public IP and the network card
- a **Standard_B1s** VM with a 30 GB Standard SSD, your SSH key (`--generate-ssh-keys` uses `~/.ssh/id_rsa.pub`, or creates one), and boot diagnostics on

It prints the `ssh arbadmin@<ip>` line when done. On first boot the VM makes its own pairing code. You can also pass `-ImageId Ubuntu2404` to get the same locked-down VM with stock Ubuntu, then install the bot as in step 5A.5.

## 6. Configure the bot and pair it with Discord

## 6.0 From first boot to paired: how it fits together

| Step | Where | What happens |
|---|---|---|
| 1. First boot | The VM | The bot is installed (quick setup) and a one-time **pairing code** is made. It's written to the Serial log and a file only the bot's account can read |
| 2. `sudo arb-setup` | SSH on the VM | You give the **token** and server ID. The bot starts and **dials out** to Discord over port 443 (HTTPS and a secure WebSocket). Nothing connects in |
| 3. The bot comes online | Discord | It shows as online in your server, but answers nothing except the pairing command |
| 4. `/dkpadmin pair` | Discord | A member with Manage Server enters the code. The bot now belongs to your server only, and the person who paired is registered as **owner** |
| 5. Owner's two-factor | Discord | The pairing reply holds the owner's two-factor key; `/dkpadmin 2fa-confirm` finishes the owner's registration and gives 10 recovery codes |

From then on, everything is done in Discord.

## 6.1 Run setup on the VM

Connect with SSH. The login banner shows the bot's state, for example `Absolute Raid Bot: NOT running (sudo arb-setup), NOT PAIRED`. Run:

```
sudo arb-setup
```

It asks for:

1. **Discord bot token.** Typing is hidden. It is written to `/etc/absolute-raid-bot/bot.env`, readable only by root, and handed to the service at start. It is never put in the image, the config file or the logs.
2. **Owner's Discord user ID** (optional).
3. **Your Discord server ID.**
4. **Staff log channel ID** (optional).

Then it starts the service and prints:

```
The bot is running.

  PAIRING CODE:  K7QM-2XVD-9FHT
  In your Discord server, a member with Manage Server runs:  /dkpadmin pair code:<the code>
```

Run `sudo arb-setup` again any time to change these; press Enter to keep a value.

## 6.2 Pair it in Discord

In your Discord server, a member with **Manage Server** runs:

```
/dkpadmin pair code:K7QM-2XVD-9FHT
```

How pairing protects you:

- The code is 12 random characters made on **your** VM. The bot keeps only a hash of it in its database, plus a file only the bot's account can read. You see it on the VM (`sudo arb-setup`, `sudo arb-status`, the login banner) or in the portal's **Serial log**, which only people with access to your Azure subscription can open.
- Until it's entered, the bot ignores every command except `/dkpadmin pair`.
- After pairing, the bot works only in that server and leaves any other server it's invited to.
- Five wrong codes and the bot throws the code away and makes a new one (see it with `sudo arb-status`).
- The code works once. After pairing the code file is deleted.

## 6.3 Finish in Discord

- **Register the owner:** the pairing reply contains the owner's two-factor key. Scan its QR code with an authenticator app (or type the key under it), run `/dkpadmin 2fa-confirm code: 123456`, and save the 10 recovery codes it shows. (If

an owner ID was set in `arb-setup` and someone else paired, the owner runs `/dkpadmin 2fa-setup` for their key.) Full steps: **Admin Registration and Account Recovery**.

- Register officers, set DKP settings and tell raiders to register: **Admin Guide**, **Admin Registration and Account Recovery**, **Raider Registration and Account Recovery**.
- From here on you don't need the VM for day-to-day work. Snapshots, raid imports, the DKP database and backups all go in and out through Discord (**Importing and Exporting Data**).

### 6.4 Save the encryption key now

```
sudo cat /opt/absolute-raid-bot/data/secret.key
```

Copy that line into your password manager. It decrypts the bot's backups and 2FA secrets. Without it a backup can't be restored on a new VM.

## 7. Network and firewall rules

Two firewalls protect the bot: the **network security group (NSG)** in Azure, in front of the VM, and **ufw** on the VM itself.

### 7.1 Azure NSG rules

| Direction | Priority | Name | Source | Destination | Port | Action | Why |
|---|---|---|---|---|---|---|---|
| Inbound | 100 | Allow-SSH-Admin | your IP /32 | any | TCP 22 | Allow | Your admin access, nobody else's |
| Inbound | 65500 | DenyAllInBound (built in) | any | any | any | Deny | Everything else is refused |
| Outbound | 100 | Allow-HTTPS-Out | VNet | Internet | TCP 443 | Allow | Discord API and gateway, Wowhead, Python packages |
| Outbound | 110 | Allow-HTTP-Out-Updates | VNet | Internet | TCP 80 | Allow | Ubuntu package and security updates |
| Outbound | 4000 | Deny-Other-Internet-Out | VNet | Internet | any | Deny | Nothing else may leave the VM |

What each outbound connection is for:

| Destination | Port | Used for |
|---|---|---|
| discord.com, gateway.discord.gg | 443 | the bot's API calls and its live connection (secure WebSocket) |
| cdn.discordapp.com, media.discordapp.net | 443 | files the bot sends or receives (PDF reports, imports) |
| www.wowhead.com, nether.wowhead.com | 443 | item names and tooltips for loot |
| pypi.org, files.pythonhosted.org | 443 | Python packages during install and updates |
| azure.archive.ubuntu.com, security.ubuntu.com | 80 / 443 | Ubuntu updates |

Notes:

- **No inbound rule is needed for Discord.** The bot opens the connection to Discord over 443 and Discord replies on that same connection.
- Azure's own services the VM needs (its DNS server 168.63.129.16 and the metadata service 169.254.169.254) are not "Internet" in NSG terms, so the deny rule doesn't block them.
- All bot traffic is encrypted with TLS. The bot checks Discord's certificate, so a man-in-the-middle can't impersonate Discord.
- Discord, Wowhead and PyPI sit behind content delivery networks whose IP addresses change, so the rules allow port 443 to the Internet service tag rather than fixed IPs. Locking to domain names needs Azure Firewall, which costs far more than the bot.
- **Keep the public IP.** Virtual networks created since March 31, 2026 are private by default; without the public IP (or a NAT gateway, which costs more) the VM has no way out and the bot goes offline.

### 7.2 ufw on the VM

`install.sh` sets:

```
ufw default deny incoming
ufw default allow outgoing
ufw limit OpenSSH
ufw --force enable
```

`limit` lets SSH in but blocks an address that tries 6 or more connections within 30 seconds. Check it any time with `sudo ufw status verbose`.

### 7.3 SSH hardening

- Keep password login off (Azure turns it off when you choose an SSH key).
- Keep the `.pem` key private. If it leaks: VM, **Help**, **Reset password**, mode **Reset SSH public key**, and give a new key.
- Change the NSG SSH source when your home IP changes; never set it to `Any`.
- Optional: Azure Bastion lets you connect from the portal with no SSH port open at all, at extra cost.

## 8. Running it 24/7

- **Starts on boot, restarts on crash.** systemd runs `absolute-raid-bot.service` with `Restart=always`; if it stops it comes back after 10 seconds. If Azure restarts the host, the bot starts when the VM boots.
- **Small and locked down.** The service runs as the `arb` account, may use at most 600 MB of memory, can't gain privileges, and can only write to its own `data` folder. The bot normally uses well under 200 MB.
- **Security updates** install automatically every day (unattended-upgrades). Some kernel updates need a reboot: run `sudo reboot` during a quiet hour now and then, or when the login banner says System restart required. The bot is back about a minute later.
- **Check on it:** `sudo arb-status` shows the service, pairing state, recent log lines, backups, memory and disk.
- **Logs:** `journalctl -u absolute-raid-bot -f` (Ctrl+C to stop).
- **Backups:** the bot makes encrypted backups (`.arbbak`) in `/opt/absolute-raid-bot/data/backups` (schedule and restore: **Importing and Exporting Data**, section 6). Keep copies **off the VM** too: after each raid night an admin runs `/dkpadmin backup` and the bot DMs them the file. A backup only opens with `secret.key` (step 6.4). Optionally turn on **Azure Backup** for the VM (VM, **Backup**) for whole-disk snapshots, at extra cost.

## 9. Common tasks

| Task | Command (on the VM) |
|---|---|
| See status and the pairing code | `sudo arb-status` |
| Change token, server or log channel | `sudo arb-setup` |

| Task | Command (on the VM) |
|---|---|
| Restart the bot | `sudo systemctl restart absolute-raid-bot` |
| Stop / start the bot | `sudo systemctl stop absolute-raid-bot` / `start` |
| The owner lost their authenticator and all recovery codes | `sudo arb-reset-2fa <Discord user ID>` |
| Move the bot to a different server | `sudo arb-pair-reset` (type YES), then pair again |

### Updating the bot

Upload the new zip as in step 5A.5, then on the VM:

```
unzip AbsoluteRaidBot-<version>.zip -d raidbot-new
cd raidbot-new
sudo bash azure/scripts/install.sh "$PWD/absolute-raid-bot"
```

The installer stops the bot, replaces the program files, keeps the `data` folder (database, audit log, backups, `secret.key`), `config.json` and the token, runs the self-tests, and starts the bot again. It stays paired, because pairing lives in the database. Take a `/dkpadmin backup` first anyway.

## 10. Troubleshooting

| Problem | What to check |
|---|---|
| SSH times out | NSG SSH rule source is your **current** IP; VM is Running; you used the public IP |
| SSH says "UNPROTECTED PRIVATE KEY FILE" | Run the `icacls` line in 5A.2 (or `chmod 600` on macOS/Linux) |
| `arb-setup` says the bot didn't start | Read the log lines it prints. "Improper token" means the token is wrong or was reset: run `sudo arb-setup` and paste the current one |
| Bot online but slash commands don't appear | Invite used the `applications.commands` scope; wait a minute, then restart Discord |
| Every command says the bot isn't paired | Run `/dkpadmin pair` with the code from `sudo arb-status` |
| "Too many wrong codes" | A new code was made: `sudo arb-status` |
| Bot offline after a while | `sudo arb-status`; check the VM isn't stopped and auto-shutdown is off; check the subscription is active (free accounts stop after 30 days unless upgraded) |
| Wowhead names missing | Outbound 443 allowed; the bot keeps working without them |
| Install fails at the self-test | Send the output to Absolute Gaming support; nothing was started |

To see the VM's console when SSH is impossible: VM, **Help**, **Boot diagnostics**, or **Serial console**.

## 11. Removing everything

When the guild no longer needs the bot, first take a final backup (`/dkpadmin backup`), then in the portal open **Resource groups**, choose `rg-raidbot`, and **Delete resource group**. That removes the VM, disk, IP and network, and the charges stop. Remove the bot from your Discord server too.

## Quick reference

- Portal: portal.azure.com. Pricing: azure.microsoft.com/pricing/calculator.
- VM: Ubuntu 24.04 LTS, Standard_B1s, 30 GB Standard SSD, public IP, auto-shutdown off.
- In: SSH (22) from your IP only. Out: TCP 443 and 80 only.
- On the VM: `sudo arb-setup`, `sudo arb-status`, `sudo arb-reset-2fa <id>`, `sudo arb-pair-reset`.
- In Discord: `/dkpadmin pair code:<code>`, then `/dkpadmin 2fa-setup`.

- **Keep safe: the `.pem` SSH key, the bot token, and `secret.key`.**