# Getting started with 2Steps

2Steps checks your applications the way a real user would: logging in, opening apps, and completing journeys, around the clock. It alerts you the moment something breaks or slows down. No code to write, and it works on anything a person can see and click: websites, Citrix, desktop apps, even mainframes. This pack covers what your team needs to plan connectivity, security, and a smooth kick-off. No prior 2Steps knowledge assumed.

## What makes 2Steps different

Most monitoring tools read an application's underlying code, so they can't see Citrix, remote desktops, or legacy systems, and they break when the code changes. 2Steps looks at the screen the way a person does.

### Wider coverage

Monitors what other tools can't see: Citrix and VDI, thick-client apps, mainframes, and logins behind MFA. If a user can see and click it, 2Steps can test it.

### Less effort

Record a journey by simply doing it. A test typically takes about 15 minutes to build, no code required, versus 7+ hours of scripting in code-based tools.

### More stability

Tests are based on what appears on screen, not the code behind it, so back-end changes are far less likely to break them.

### Clearer answers

Every run is recorded on video. When something fails, you watch exactly what happened, the way a user would have seen it.

**A simple analogy:** a code-based tool checks a room by reading the architect's blueprints. It only works if the blueprints exist and are up to date. 2Steps walks in and looks around. It sees what's really there, the way your users do.

CONNECTIVITY

# Where 2Steps sits, and what it needs to reach

2Steps runs on a small Linux virtual machine inside your network. From there it needs two kinds of connectivity: your team's browsers reaching the 2Steps dashboard, and the 2Steps server reaching the applications it monitors.

## Two ways to connect to your applications

### 1 · Direct connection

The 2Steps server connects straight to each application. Simple to picture, but **every new application needs new firewall rules**, which adds delay each time you want to monitor something new.

### 2 · Via your web proxy RECOMMENDED

The 2Steps server uses the same web proxy your staff already use. One firewall change up front. Then **new tests need no further firewall work**.

## Firewall rules to plan for

| FROM → TO | WHY |
|---|---|
| Staff browsers → 2Steps server (HTTPS) | So your team can log in, build tests, and view results. |
| 2Steps server → apps or web proxy | So tests can reach the applications being monitored. |
| 2Steps server → Windows desktops (RDP/VNC) | Only if you monitor Windows, Citrix, or thick-client apps. 2Steps drives a real desktop to test them. |
| 2Steps servers ↔ each other (SSH) | Only for multi-site setups, where remote "spokes" report back to a central "hub". |

SERVER YOU'LL NEED

One virtual machine: **4 vCPU · 4 GB RAM · 50 GB disk**, running RHEL 9/10 or a compatible Linux (e.g. Rocky Linux). Installation takes roughly an hour per server.

GOOD TO KNOW

Submit the firewall plan to your network team early. It's usually the longest lead-time item in the whole project.

SECURITY

# Set up safely from day one

Three things keep a 2Steps installation secure: encrypting logins, giving test accounts only the access they need, and managing the video recordings sensibly.

1. **Encrypt logins with SSL (HTTPS)**
   Usernames and passwords travel between browsers and the 2Steps server, so every installation should use an SSL certificate. You'll need a domain name for the server (e.g. *2steps.yourorganisation.com*) and a certificate from whichever Certificate Authority your organisation uses; your IT or security team will know. This is a standard request they handle all the time.

2. **Service accounts with least privilege**
   Each test logs in using a dedicated service account. The rule is simple: give each account **only the permissions its workflow needs**: if a test opens application XYZ, its account needs access to XYZ and nothing more. Your security team will appreciate that this is the default approach, not an afterthought.

3. **Manage video recordings**
   2Steps records a video of every test run. That's what makes failures so quick to diagnose. Videos stay on your own server (nothing leaves your network), stored in a dedicated disk area so they can't fill up the machine. Set a retention policy during install so old videos are cleared out automatically.

THE QUESTION SECURITY TEAMS ALWAYS ASK

**"Why does the service account need desktop access?"** Because 2Steps tests applications visually: it logs into a real Windows desktop and interacts with what's on screen, just like a person. Accounts used for Windows-based tests therefore need permission to log into the desktop, not just the application. Worth explaining up front; it saves a round of back-and-forth later.

**MFA is not a blocker.** Unlike code-based tools, 2Steps can complete logins protected by MFA / 2FA, so you don't need to weaken authentication on monitored systems to test them.

PLANNING

# Deployment checklist

Work through these in order. Most items are hours of effort, not days. The lead times sit with the teams you depend on, so make the requests early.

| STEP | WHO | EFFORT |
|---|---|---|
| ☐ **Set the scope**: standalone or Splunk-connected; list the user journeys to monitor | Project lead + management | 2 hrs |
| ☐ **Size the system**: how often each journey runs, roughly how long it takes | Monitoring team | 1 hr |
| ☐ **Request the VM(s)**: Linux server(s), plus Windows desktops if needed | Infrastructure team | 1 day |
| ☐ **Submit the firewall plan**: connections from page 2, approved by the network team | Network team | Varies |
| ☐ **Install 2Steps**: run the installer, add SSL certificates | Monitoring team (+ partner) | ~1 hr / server |
| ☐ **Create service accounts**: least-privilege accounts for each workflow | Security + accounts team | Varies |
| ☐ **Document the workflows**: screenshots of each journey, step by step | Application teams | 2 hrs |
| ☐ **Build and schedule tests**: record journeys, set schedules and alerts | Monitoring team | ~15 min / test |

## Who to involve

**Project lead**
Owns scope, timeline, and chasing the dependencies below.

**Infrastructure team**
Creates the Linux VM(s) and any Windows desktops.

**Network team**
Approves and applies the firewall rules.

**Security team**
Signs off service accounts, SSL, and permissions.

**Application teams**
Confirm the workflows and flag any that change data.

**Monitoring team**
Runs 2Steps day to day: builds tests, watches results.

KICK-OFF

# Your first 30 days

| | |
|---|---|
| **Week 1** | **Scope and requests.** Hold the kick-off, agree standalone vs Splunk, and pick 3–5 high-value journeys to start with. Submit the VM and firewall requests the same week. They're the long poles. |
| **Week 2** | **Install and connect.** Install 2Steps (about an hour per server), add SSL, and connect any Windows desktops. Request the service accounts while this happens. |
| **Week 3** | **Build the first tests.** Record the chosen journeys (about 15 minutes each), put them on a schedule, and set up alerts so the right people hear about failures. |
| **Week 4** | **Review and expand.** Watch a week of results and video replays, tune alert thresholds, then agree the next batch of journeys to bring on. |

## Which tests to build first

Start with the journeys that would hurt most if they broke: usually a login to a business-critical app, then one complete everyday task within it. Keep each test focused on a single workflow: small tests are easier to maintain and quicker to diagnose.

## Three pitfalls to avoid

**Data-changing tests**
A test that creates records runs 24/7, so check the application can handle that before scheduling it.

**Wrong timezone**
Schedules follow the server's clock, so set the timezone at install or tests will run at the wrong times.

**Over-scheduling**
Each scheduled run uses a license (every-10-minutes = 6 per hour). Match frequency to how critical the journey is.

**You're not doing this alone.** 2Steps support and your implementation partner will walk your team through every documented step, and free self-paced training is available at 2Steps Education.

**education.2steps.io**
**support@2steps.io**