JANZEER BLOCKCHAIN

دليل مشغّل العقدة

التثبيت · التشغيل · التحقّق

إصدار الوثيقة 1.0 · شبكة جنزير · 2026

دليل مشغّل عقدة Janzeer

هذا الدليل موجَّه للأشخاص الذين يُشغِّلون عُقَد Janzeer ولمن يرغبون في أن يصبحوا مُصادِقين. يغطي ما تفعله العقدة، وكيفية **تشغيل عقدة مُصادِق من ثنائي أصلي مُسبَق البناء (prebuilt native binary)** (مسار المستخدم النهائي)، والإعداد (المنافذ، والأسرار، وNTP، وقاعدة البيانات، وتجاوزات الكتلة التكوينية)، وكيفية التسجيل كمُصادِق (بلا إذن مسبق، خلف وديعة غير قابلة للاسترداد) باستخدام المحفظة غير الحافظة، وكيفية مراقبة العقدة وتشغيلها — و، للفريق الداخلي، كيفية البناء والتشغيل من المصدر. إن Janzeer هي طبقة أولى (Layer-1) **بلا حصّة وبلا إذن مسبق** ذات معروض ثابت — يُسجِّل المُصادِقون بلا تصويت وبلا موافقة، ويُنتجون الكتل في تناوب ثابت، ويُنهون بإجماع BFT ثنائي المرحلة؛ والعقدة **غير حافظة** — فهي تتحقق من التواقيع ولا تحتفظ أبدًا بمفاتيح المستخدمين. وللاطّلاع على التفاصيل الاقتصادية والبروتوكولية المُشار إليها هنا، راجع الورقة البيضاء والورقة الصفراء.

> **Janzeer ليست مفتوحة المصدر، لكن العقدة تنزيل عام.** لا يبني المستخدمون النهائيون من المصدر ولا يُشغِّلون ملف jar: بل يُنزِّلون الحزمة الأصلية لمنصّتهم (Linux أو Windows أو macOS، أو صورة Docker)، ويتحققون من مجموعها الاختباري، ويُشغِّلونها (§2). أمّا مادة البناء من المصدر (Gradle، وjar، و`nodes. لا يوجد ملف JAR ولا بناء من المصدر — ويسرد §6 ما يُوزَّع.

1. نظرة عامة: ما الذي تفعله العقدة

تؤدي كل عقدة Janzeer العمل الأساسي ذاته: تتصل بالنظائر عبر رابط P2P موثَّق، وتنشر الكتل والمعاملات وتتحقق منها، وتُزامِن السلسلة، وتحفظ الحالة في قاعدة بيانات، وتُقدِّم واجهة REST البرمجية ونقاط نهاية المستكشف. وهناك دوران تشغيليان، والفرق بينهما يتعلق كليًا بما إذا كانت العقدة تحتفظ بمفتاح توقيع صالح.

- **عقدة مراقبة فقط (مزامنة فقط)** — أي عقدة مفتاحها **غير مُسجَّل** على السلسلة. تُقلِع وتتصل بالنظائر وتُزامِن السلسلة، وتقبل المعاملات في ذاكرة المعاملات المعلّقة (mempool) الخاصة بها، وتُقدِّم واجهة القراءة البرمجية كاملةً والمستكشف. وهي **لا** تُنتج كتلًا **ولا** تُدلي بأصوات النهائية. هذا ما يكون عليه التنزيل الجديد، وهو الوضع الافتراضي لأي شخص يريد فقط عقدة مُزامَنة أو خلفية لمستكشف/واجهة برمجية. ولا تحتاج إلى أي منفذ وارد.
- **مُصادِق نشط** — عقدة يكون مفتاح المُصادِق الخاص بها (المفتاح العام لـ `secret` في `config.json`) مُسجَّلًا وضمن **المجموعة النشطة**. تمتلك فترات إنتاج الكتل (تناوب دائري حتمي) وتُدلي بأصوات `COMMIT` / `PREPARE` موقَّعة. ولكي تكون نشطة، يجب أن يكون المفتاح إمّا **مِرساة** (anchor) تكوينية أو مُصادِقًا مُسجَّلًا دخل المجموعة النشطة — والاختيار يجري حصريًّا حسب **ترتيب التسجيل** (الأقدم تسجيلًا أولًا)، لا بالأصوات ولا بالتقييم (انظر §4).

تذكير بالغ الأهمية: **لا يمكن لعقدة مفردة أن تُنهي أي شيء.** النهائية ثنائية المرحلة من نوع (`COMMIT` → `PREPARE`) BFT وتتطلّب **نِصابًا** حيًّا من المجموعة النشطة — `bftQuorum(N) = floor(2N/3) + 1` على حجم المجموعة النشطة الفعلي `N`. ومع المراسي التكوينية الثلاث، يكون النِّصاب 3، لذا يجب أن تكون الثلاث جميعها حيّة ومُصوِّتة لإنهاء الكتل. أمّا العقدة المنفردة فستقبل المعاملات في ذاكرة معاملاتها المعلّقة لكنها لن تُثبِّت كتلةً بمفردها أبدًا. شغِّل **ما لا يقلّ عن 4 مُصادِقين** كي تتحمّل الشبكة عطلًا واحدًا (`N ≥ 3f+1`).

2. تشغيل عقدة (التنزيل العام)

يمكن لأي شخص تشغيل عقدة Janzeer من **التنزيل الثنائي العام** — بلا Java، وبلا خادم قاعدة بيانات، وبلا بناء. أرشيف واحد لكل منصّة (Linux x86-64، وLinux arm64 لأجهزة راسبيري باي، وWindows x86-64، وmacOS بمعالجات Apple silicon / Intel) بالإضافة إلى صورة Docker، وكلها مبنيّة من الإصدار نفسه: ملف تنفيذي أصلي، ومكتباته الزمنية، وملف `genesis.json` العام للشبكة، ومُشغِّل، و`README.txt`، و`LICENSE.txt`، و`VERSION`، و`SHA256SUMS`. **لا تحتوي الحزمة على أي مفاتيح أو أسرار**؛ فعقدتك تُنشئ هويتها الخاصة عند أول تشغيل.

هناك طريقتان بالضبط لتشغيلها، وبالثنائي **نفسه**:

الدور	ما الذي يفعله	ما يحتاجه
مراقبة فقط (الافتراضي)	يتابع السلسلة، ويُقدِّم المستكشف وواجهة REST/JSON-RPC/WebSocket على جهازك، ولا يُنتج ولا يُصوِّت أبدًا.	لا شيء سوى اتصال إنترنت صادر — فالعقدة تتصل بنظائرها من تلقاء نفسها.
مُصادِق	العقدة نفسها بعد تسجيل مفتاحها على السلسلة: تُنتج الكتل في فتراتها وتُصوِّت على النهائية.	منفذ P2P الخاص بها قابلًا للوصول من الإنترنت، ومحفظة تحمل **وديعة JNZ 2000 غير قابلة للاسترداد** + رسم JNZ 3، وعقدة واحدة بالضبط لكل مفتاح.

لا يوجد تفويض ولا حصّة (staking): إمّا أن تُشغِّل عقدة أو لا.

الخطوة 1 — التنزيل والتحقق

احصل على أرشيف منصّتك من صفحة **المُصادِقين** على janzeer.org (أو من العنوان الوارد في `release_node.md` للفريق). تعرض الصفحة قيمة SHA-256 لكل أرشيف؛ تحقَّق منها قبل فك الضغط:

```
# Linux / macOS
sha256sum -c SHA256SUMS          # أو: shasum -a 256 -c SHA256SUMS
tar -xzf janzeer-node-<version>-<network>-linux-x64.tar.gz
# Windows (PowerShell)
Get-FileHash janzeer-node-<version>-<network>-windows-x64.zip   # قارِن بالقيمة المنشورة
Expand-Archive janzeer-node-<version>-<network>-windows-x64.zip
```

`<network>` هي `mainnet` (الشبكة العامة الوحيدة). ويمكن للعدة نفسها أن تنتج حزمة `testnet` (معرِّف الشبكة `janzeer-testnet`) لشبكة اختبار خاصة تشغّلها بنفسك — عملاتها بلا قيمة؛ ولا توجد شبكة اختبار عامة. لا يمكن لحزمة شبكة أن تنضم إلى شبكة أخرى أبدًا: فمعرِّف الشبكة يُوقَّع في كل معاملة ويُثبَّت في مصافحة النظائر.

المتطلبات: نظام تشغيل 64-بت، و1 جيجابايت من الذاكرة الحرّة، و20 جيجابايت من القرص، وساعة **مُزامَنة عبر NTP** (يتوقف المُصادِق عن الإنتاج إذا انحرفت ساعته أكثر من 250 مللي ثانية).

الخطوة 2 — التشغيل

```
./start.sh          # Linux / macOS
start.cmd           # Windows
docker run -d --name janzeer -p 127.0.0.1:7019:7019 -p 9199:9199 -v janzeer-
data:/opt/janzeer/data ghcr.io/janzeerorg/janzeer-node:latest   # Docker (x86-64 و
arm64)
```

تُنشئ العقدة `data/.` (دفتر الأستاذ + `data/config.json`، أي هويتها)، وتتصل بالنظائر البذرية المُضمَّنة في المُشغِّل، وتُزامِن السلسلة، وتُقدِّم المستكشف على `http://localhost:7019`. تحقَّق من التقدّم بـ:

```
curl -s http://localhost:7019/api/v1/info
# …
"networkId":"janzeer","genesisHash":"…","version":"0.1.14","syncStatus":"SYNCHRONIZED"
…
```

تصبح `syncStatus` بقيمة `SYNCHRONIZED` عندما تلحق العقدة بالسلسلة. هذه عقدة **مراقبة فقط** مكتملة.

كل شيء يُضبط عبر متغيّرات بيئة يقرؤها المُشغِّل (انظر `README.txt` داخل الحزمة): (7019) `JANZEER_REST_PORT`، و(9199) `JANZEER_P2P_PORT`، و`JANZEER_ROOT_NODES` (النظائر البذرية)، و`JANZEER_DATA_DIR` (`./data`)، و`JANZEER_JAVA_OPTS` (`-Xmx256m`)، و`DB_PASSWORD` / `DB_USERNAME` / `DB_URL` لاستخدام PostgreSQL بدلًا من قاعدة البيانات المُضمَّنة. تأتي حزمة Linux مع وحدة `janzeer-node.service` وتعليمات التثبيت داخلها.

الخطوة 3 — النسخ الاحتياطي للهوية

يحمل `data/config.json` مفتاح توقيع العقدة (**مفتاح المُصادِق** الذي ستُسجِّله) وعبارة استرداد محفظة العقدة الخاصة. انسخه إلى مكان آمن وأبقِه سريًّا. أمّا بقية `/data` فهي دفتر الأستاذ ويمكن دائمًا إعادة مزامنتها من الشبكة.

الخطوة 4 — كن مُصادِقًا

1. **افتح منفذ P2P** (الافتراضي `tcp/9199`) على الإنترنت كي تستطيع العقد الأخرى الاتصال بعقدتك. يعمل المدقق خلف جهاز توجيه منزلي من دونه أيضًا: كل كتلة وتصويت وشهادة تنتقل عبر الاتصالات الصادرة التي تحتفظ بها عقدتك مع نظائرها، وكل عقدة تعيد إرسال ما تستلمه. يضيف المنفذ المفتوح روابط مباشرة مع المدققين الآخرين ويتيح للنظائر الوصول إليك عند انقطاع اتصالاتك، فافتحه حين تستطيع.
2. **اقرأ مفتاح عقدتك:** `curl -s http://localhost:7019/api/v1/info` → `"nodeKey": "02…"` (يظهر أيضًا في المستكشف الذي تُقدِّمه عقدتك).
3. **من محفظة Janzeer** (محفظة الويب في أي مستكشف، أو تطبيق الهاتف): افتح تبويب **المُصادِقين**، والصق مفتاح العقدة، واختر **تسجيل**. توقِّع المحفظة معاملة التسجيل وتُرسلها (`POST /api/v1/transactions/validators`) حاملةً **وديعة 2000 JNZ غير القابلة للاسترداد** بالإضافة إلى الرسم الدقيق 3 JNZ. لا تغادر المفاتيح المحفظة أبدًا؛ فالعقدة غير حافظة.
4. تُقبَل عقدتك في المجموعة النشطة عند **حدود الحقبة التالية** وتبدأ الإنتاج في فتراتها. تأكَّد عبر `api/v1/validators/active/`.

من هنا فصاعدًا ينطبق §4: تذهب المكافآت بنسبة 100% إلى محفظة المُصادِق المُنتِج، ويمكنك الخروج بلياقة (`POST /api/v1/transactions/exit-validators`؛ وتبقى الوديعة غير قابلة للاسترداد)، وتُطرَد العقدة الصامتة في فترتها بسبب التعطّل، ويجب أن تُشغِّل **عقدة واحدة بالضبط لكل مفتاح** — فعقدتان تتشاركان `data/config.json` تُعاقبان (slashing) بسبب التوقيع المزدوج.

الخطوة 5 — الترقية

تُخبَر أنت وتقرّر أنت. تتحقق العقدة مرة كل يوم من وجود إصدار أحدث (`node.update.url`، وهو بيان المشروع على `downloads.janzeer.org/node/latest.json`؛ إشعار فقط، فهي لا تنزّل شيئًا ولا تثبّته؛ الخيار `--node.update.enabled=false` يعطّل ذلك). يظهر الإصدار الأحدث سطر تحذير واحدًا في السجل، وإشعارًا في أعلى المستكشف، وفي `GET /api/v1/info/update`.

تأتي حزم لينكس وماك مع `update.sh`: الأمر `update.sh --check/.` يبلّغ فقط، والأمر `update.sh/.` ينزّل الحزمة الجديدة، ويتحقق من **توقيع minisign** الخاص بالمشروع على ملف `SHA256SUMS` للإصدار ومن مجموع الأرشيف مقابله، ويبقي `data/.`، ويستبدل الملفات، ويعيد تشغيل `janzeer-node.service` إن كانت العقدة تعمل خدمةً (والأمر `update.sh --/.` يعيد الإصدار السابق من `rollback/previous.`). يحتاج إلى تثبيت `minisign` ويرفض أي إصدار غير موقّع أو معدَّل. المفتاح العام للإصدارات (minisign، معرّف المفتاح `9F798583B345AE01`) هو `RWQBrkWzg4V5n5wUBa9GNPpa95Z7WFT2EY5QrabMv/DP+KG+gJsPIgdr`؛ وهو مطبوع داخل `update.sh` وفي `README.txt` داخل الحزمة وعلى https://janzeer.org/validators، فقارنوا بينها. للتحقق من إصدار يدويًا: `minisign -Vm SHA256SUMS -P RWQBrkWzg4V5n5wUBa9GNPpa95Z7WFT2EY5QrabMv/DP+KG+gJsPIgdr`. ويندوز: أوقف العقدة، وفكّ الملف المضغوط الجديد فوق المجلد (أبقِ `data\`)، وتحقّق من `SHA256SUMS`، ثم شغّلها. آرتش لينكس: `pacman -U` للحزمة الجديدة.

منذ الإصدار 0.1.11 تحمل كل حزمة أيضًا `jz`، عميل سطر الأوامر، بجوار الملف التنفيذي `janzeer`: `./jz node status` (الإصدار والمزامنة والصحة وتوفر التحديث)، و`./jz node update` (الخروج 10 = يوجد تحديث، و`--run` يثبّته)، و`./jz watch blocks`، و`./jz info`؛ ولا يحتاج إلى `--node-dir` عند تشغيله من مجلد الحزمة. الدليل الكامل هو دليل عميل سطر الأوامر (`cli.md`)؛ ويبقيه `update.sh` محدّثًا مع العقدة.

يدويًا على أي منصة: أوقِف العقدة، وفكّ ضغط الحزمة الجديدة فوق المجلد (أبقِ `data/.`)، وتحقَّق من `SHA256SUMS`، ثم شغِّلها من جديد. يذكر `VERSION` إصدار بروتوكول الاتصال؛ والنظائر التي تعمل بإصدار بروتوكول آخر تقطع الاتصال بك مع `protocol version mismatch` في السجل — وهذا السطر يعني «قم بالترقية». يُرقّي المُصادِقون واحدًا تلو الآخر كي تحافظ الشبكة على نِصابها، إلا حين تذكر ملاحظات الإصدار أن جميع العُقَد تُعاد تشغيلها معًا (رفع إصدار البروتوكول).

استكشاف الأخطاء

سطر السجل / العَرَض	المعنى	الحلّ
`GENESIS MISMATCH: …` ثم تخرج العملية	يصف `genesis.json` الموجود في المجلد سلسلةً أخرى غير دفتر الأستاذ في `data/.` (مثلًا حزمة شبكة اختبار فوق دليل بيانات للشبكة الرئيسية) أو أن الملف عُدِّل.	استخدم الحزمة المطابقة، أو انقل `data/.` بعيدًا وأعِد المزامنة.
`genesis file … is for network 'janzeer-testnet' but this node runs consensus.network-id ''janzeer`	مُشغِّل مُعدَّل يدويًا يوجِّه عقدة شبكة رئيسية إلى كتلة تكوينية لشبكة اختبار (أو العكس).	أعِد استخراج الحزمة؛ ولا تخلط الملفات بين الشبكات.
`Closing … after a decode error: protocol version mismatch`	عقدتك أو النظير يعمل بإصدار أقدم.	قم بالترقية.
`Rejecting … chain-spec digest mismatch` في سجل نظير	قواعد إجماع مختلفة أو كتلة تكوينية مختلفة — سلسلة أخرى.	كما أعلاه: الحزمة المطابقة.
`syncStatus` تبقى `PROCESSING`	ما زالت تلحق بالسلسلة، أو لا يوجد نظير بذري قابل للوصول.	انتظر؛ وتحقَّق من `JANZEER_ROOT_NODES` والاتصال الصادر.
مُسجَّلة لكنها لا تُنتج	الساعة منحرفة أكثر من 250 مللي ثانية، أو منفذ P2P مغلق، أو لم تمرّ حدود الحقبة بعد.	أصلِح NTP / جدار الحماية؛ وانتظر الحقبة التالية.

3. الإعداد

يأتي الإعداد من القيم الافتراضية المُضمَّنة، وتجاوزات سطر الأوامر `--key=value` (يمرِّرها المُشغِّل)، ومتغيّرات البيئة، وملف الهوية `config.json` الخاص بكل عقدة. على حزمة مُنزَّلة لا تلمس عادةً شيئًا: فمتغيّرات بيئة المُشغِّل (§2، الخطوة 2) تغطي المنافذ والنظائر البذرية ودليل البيانات وقاعدة البيانات.

المنافذ

الخاصية	القيمة الافتراضية	الغرض
`server.port`	`7019`	واجهة REST البرمجية / واجهة المستخدم
`node.port`	`9199`	P2P (Netty TCP)

`*.node` — الشبكة والساعة

الخاصية	القيمة الافتراضية	الغرض
`node.root-nodes`	النظائر البذرية للشبكة (مُضمَّنة في مُشغِّل الحزمة؛ ويتجاوزها `JANZEER_ROOT_NODES`)	النظائر البذرية/الجذرية التي يُتَّصل بها عند بدء التشغيل للاكتشاف.
`node.peers-number`	`3`	العدد المستهدف لاتصالات النظائر الواردة.
`node.protocol-version`	`3.4.0`	إصدار بروتوكول اتصال P2P؛ النظير الذي يعمل بإصدار آخر يُقطَع اتصاله (`protocol version mismatch` في السجل).
`node.ntp-servers`	(قائمة مُعدَّة)	خوادم NTP المستخدمة لمزامنة الساعة.

مزامنة الساعة ضرورية للإنتاج. توقِف العقدة إنتاج الكتل إذا لم تكن ساعتها مُزامَنة (يجب أن يكون انحراف NTP ضمن `ntp-offset-threshold = 250 ms` — وقد شُدِّد ليلائم الفترات الزمنية بساعة الحائط البالغة 15 ثانية) أو إذا لم تكن مُزامَنة بالكامل. والعقدة ذات الساعة المنحرفة ستظل تُقدِّم الواجهة البرمجية وتُزامِن، لكنها ستفوِّت فترات إنتاجها بصمت — حافظ على عمل NTP.

`*.consensus` — المعاملات الاقتصادية ومعاملات الإجماع (أبرزها)

هذه المعاملات هي **مجموعة القواعد المثبَّتة عند التكوين**: فكل ما يقع تحت `*.consensus` هو مجموعة قواعد الإجماع القانونية المثبَّتة عند الكتلة التكوينية. تحسب كل عقدة **بصمة مواصفات السلسلة** (double-SHA256 على هذه المعاملات) وتتبادلها في مصافحة P2P؛ ويُرفض أي نظير تختلف بصمته. **لا تُغيِّر أي قيمة من `*.consensus` على شبكة حيّة** — فالعقدة التي تُعدِّل واحدة منها تنشقّ عن الشبكة بدلًا من الانضمام إليها. الإعدادات التشغيلية وحدها (المنافذ، وقاعدة البيانات، والنظائر، وNTP) هي ما يختلف بين العُقَد. ومع ذلك ينبغي أن تعرف هذه المعاملات لتُشغِّل بشكل صحيح؛ راجع الورقة الصفراء للمجموعة الكاملة.

الخاصية	القيمة الافتراضية	المعنى
`consensus.network-id`	`janzeer`	هوية الشبكة، تُوقَّع في كل معاملة وفي المصافحة. تستخدم شبكة الاختبار الخاصة `janzeer-testnet`.
`consensus.promoters-count`	`21`	الحجم الأقصى للمجموعة النشطة.
`consensus.rotating-producer-seats`	`5`	المقاعد المُنتخَبة التي تتناوب عبر دكّة الاحتياط.
`consensus.producer-candidate-pool-size`	`100`	حجم دكّة الاحتياط للتناوب المُنصِف.
`consensus.epoch-height`	`17000`	الكتل الرئيسية في كل حقبة (≈ 3 أيام؛ إعادة انتخاب المجموعة النشطة في كل حقبة).
`consensus.time-slot-duration`	`ms 12000`	نافذة الإنتاج النشطة لكل فترة زمنية.
`consensus.time-slot-interval`	`ms 3000`	الفاصل بين الفترات الزمنية (الفترة الكاملة = كتلة كل 15 ثانية).
`consensus.amount-promoter-tx`	`2000`	وديعة التسجيل غير القابلة للاسترداد (← الخزينة).
`consensus.fee-promoter-tx`	`3`	الرسم الدقيق لمعاملة تسجيل المُصادِق.
`consensus.safety-floor-validators`	`4`	لا تُلغى المراسي تدريجيًا أبدًا إلى ما دون هذا العدد من المُصادِقين السليمين.
`consensus.anchor-demotions-per-epoch`	`1`	تتخلّى المراسي المميَّزة عن واحدة في كل حقبة كلما أثبت المستقلّون جدارتهم.
`consensus.validator-proven-epochs`	`2`	الحقب المتتالية التي يجب أن يُنتج فيها غير-المِرساة ليُحتسَب مُثبَتًا.
`consensus.equivocation-ban-threshold`	`2`	المخالفات المُثبَتة المتمايزة ← حظر دائم.
`consensus.slash-reporter-bounty`	`5`	المكافأة المدفوعة لمُبلِّغ المصادرة (من الخزينة).

> **الأسماء القديمة "promoter".** تقول الواجهة البرمجية وهذا الدليل **مُصادِق (validator)**، لكن بعض المُعرِّفات الداخلية تُبقي على الكلمة الأصلية **promoter** — وهي المفهوم ذاته باسم قديم. وهذه لم تتغيَّر: مفاتيح الإعداد `consensus.promoters-count` / `consensus.amount-promoter-tx` / `consensus.fee-promoter-tx` / `count`، وأنواع المعاملات/الحالة `PromoterState` / `ExitPromoterTx` / `PromoterTx`، وجداول قاعدة البيانات `*_promoter`.

قاعدة البيانات (PostgreSQL)

افتراضيًّا تستخدم العقدة H2 المُضمَّنة (`jdbc:h2:file:./data/janzeer;MODE=PostgreSQL`)، مخزَّنةً تحت دليل `/data` الخاص بالحزمة. ولاستخدام PostgreSQL، عيِّن:

```
export DB_URL=jdbc:postgresql://localhost:5432/janzeer
export DB_USERNAME=janzeer
export DB_PASSWORD=********
```

يُشحَن كلا المُشغِّلَين (drivers) مع العقدة، وخط أساس مخطط Flyway مستقل عن اللهجة، لذا لا حاجة إلى أي تغييرات في الشيفرة.

`config.json` — هوية العقدة

يحمل `config.json` هوية العقدة (يمكن تعيين موقعه بـ `--config-path`؛ ويشير المُشغِّل الأصلي إلى `./config.json`). وإذا كان الملف مفقودًا، تُنشئه العقدة تلقائيًا عند الإقلاع الأول:

```
{ "externalPort": 9199, "secret": "<node-signing-key>" }
```

- **`secret`** — مفتاح توقيع الكتل الخاص بالعقدة؛ ومفتاحه العام هو **مفتاح مُصادِق** العقدة. بوجود `secret` صالحة، تُوقِّع العقدة الكتل والأصوات (ويجب أن تكون مِرساة أو مُصادِقًا مُنتخَبًا لتكون نشطة). **إنّ `secret` فارغة أو مفقودة تعني المزامنة فقط: تُقلِع العقدة وتتصل بالنظائر وتُزامِن، لكنها لا تستطيع الإنتاج أو التصويت.**
- **المحفظة** — محفظة العقدة الخاصة، المُشتَقّة إلى جانب الهوية.
- **`externalPort`** — منفذ P2P المُعلَن للنظائر.

عامِل `config.json` معاملة السرّ وخُذ منه نسخة احتياطية — فهو يحتوي على مفتاح التوقيع وهو هوية العقدة.

تجاوز الكتلة التكوينية وصنبور صندوق التطوير — لشبكة الاختبار فقط

يتيح لك مِفتاحان بدء سلسلة جديدة أو تمويل محفظة مسبقًا للاختبار. **كلاهما للتطوير/شبكة الاختبار فقط — لا تُفعِّلهما أبدًا في الإنتاج**، وهما وثيقا الصلة فقط عند إقامة شبكتك الخاصة (§6)، لا عند الانضمام إلى شبكة قائمة.

- **صنبور صندوق التطوير** — عيِّن `JANZEER_DEV_FUND=0xaddress:amount` (خاصية Spring `janzeer.dev-fund`) لبَذْر حساب مموَّل واحد في الكتلة التكوينية، بحيث يمكنك اختبار إرسال العملات دون كسبها أولًا. **مُعطَّل افتراضيًّا. لا تُعيِّنه أبدًا في الإنتاج.**
- **تجاوز الكتلة التكوينية** — الكتلة التكوينية مدفوعة بالإعداد: تتيح `janzeer.genesis.anchors` و`janzeer.genesis.signer-secret`، وعناوين الحفظ الاختيارية / `janzeer.genesis.founder`، `janzeer.genesis.distribution`، للعقدة بَذْر كتلة تكوينية جديدة **أثناء التشغيل** دون تعديل المصدر. وإذا كانت `.distribution` / `founder` غائبةً، تنطوي تلك المخصَّصات في الخزينة عديمة المفتاح، فلا يُسَكّ شيء يتجاوز السقف الثابت البالغ 100,000,000.

إنّ أسرار المراسي المُلتزَم بها والقيم الافتراضية للكتلة التكوينية هي **قيم تطوير/شبكة اختبار** — أبدِلها بمفاتيح جديدة لأي نشر حقيقي.

4. أن تصبح مُصادِقًا

التسجيل **بلا إذن مسبق** — لا تصويت ولا موافقة ولا قفل للأموال. تصبح مُصادِقًا بتشغيل عقدة بمفتاح توقيع صالح (§2) وتقديم معاملة واحدة على السلسلة من المحفظة غير الحافظة؛ فالتسجيل وحده يُدخِلك. أمّا الشرح الكامل من البداية إلى النهاية (المحفظة، والوديعة، وعلامة تبويب **المُصادِق**) فهو في §2، الخطوتين 4–5. ويشرح هذا القسم كيف يتصرّف الدور بعد تسجيلك.

1. **سجِّل** من المحفظة التي ستتلقّى المكافآت. تُكلِّف معاملة التسجيل (`POST /api/v1/transactions/validators`، حاملةً `validatorKey` عقدتك) **رسمًا دقيقًا قدره 3** بالإضافة إلى **وديعة غير قابلة للاسترداد قدرها 2000** (`amount-promoter-tx`، تُدفَع إلى بنك الخزينة). ويجب أن يكون `validatorKey` فريدًا. إنّ الوديعة، مع مقعد واحد لكل مفتاح، هي واقي Sybil — وهي **لا** تُعاد عند خروجك.

2. **ادخل المجموعة النشطة بترتيب التسجيل.** الاختيار يجري حسب **ترتيب التسجيل**، لا بالأصوات ولا بالتقييم: يُرتَّب المُصادِقون المؤهَّلون حسب زمن تسجيلهم (`createDate`، الأقدم تسجيلًا أولًا). وما دام عدد المُصادِقين المُسجَّلين عند سقف المقاعد أو دونه، فجميعهم نشطون؛ ويتناوب الفائض بإنصاف. المجموعة النشطة هي **المراسي** التي ما زالت مميَّزة بالإضافة إلى أقدم المُصادِقين المُنتخَبين تسجيلًا، مملوءةً حتى `promoters-count = 21`. ومن هؤلاء، تتناوب `rotating-producer-seats = 5` من المقاعد المُنتخَبة بالتناوب الدائري (تناوب مُنصِف، مُرتَّب حسب التسجيل) عبر دكّة احتياط تصل إلى `producer-candidate-pool-size = 100`.
3. **اخرج بلياقة** عندما تريد الاعتزال. قدِّم معاملة خروج المُصادِق (`POST /api/v1/transactions/exit-validators`، رسم ≤ 0.01) بمفتاحك `validatorKey`؛ فتُزال من المجموعة النشطة ومن جدول الفترات الزمنية عند حدود الحقبة التالية، بحيث تتوقف عن تلقّي فترات على الفور بدلًا من الإضرار بالحيوية إلى أن يطردك التعطّل. وتبقى وديعة التسجيل مفقودةً.

> **منحة المُصادِق (منذ 2026-10-10).** يحصل أول 10 مشغّلين من خارج المشروع يشغّلون عقدة على الشبكة الرئيسية متزامنة ويمكن الوصول إليها لمدة 7 أيام على وديعة التسجيل (مع الرسم) من محفظة النظام البيئي. القواعد والتقديم: https://janzeer.org/validators#grant.

المكافآت

عندما تُنتج عقدتك كتلة، تتلقّى المحفظة المُنتِجة **100%** من مكافأة الكتلة — لا تفويض، ولا اقتسام عمولة، ولا مطالبة بالمكافآت. معاملة المكافأة مُولَّدة من النظام (رسم 0، nonce 0) ويُتحقَّق منها مقابل القيمة القانونية على كل عقدة. ومن كل رسم معاملة، تذهب **80% إلى محفظة مُنتِج الكتلة** وتعود **20% إلى بنك الخزينة** (`treasury-fee-share-bps = 2000`)؛ ولا تُحرَق الرسوم أبدًا.

الإلغاء التدريجي للمراسي تلقائيًّا

تبدأ المراسي مميَّزةً (دائمًا ضمن المجموعة، ومحصَّنة ضد التعطّل) وتتخلّى عن هذا الامتياز **واحدة في كل حقبة** (`anchor-demotions-per-epoch = 1`) بمجرد وجود عدد كافٍ من المُصادِقين **المستقلّين المُثبَتين** — أي غير-المراسي الذين أنتجوا في كل حقبة من الحقب الأخيرة البالغة `validator-proven-epochs = 2` — دون أن تُخفِض أبدًا المجموعة السليمة إلى ما دون `safety-floor-validators = 4`. وإذا انهار المستقلّون لاحقًا إلى ما دون هذا الحد، تعود المراسي الخاملة للانخراط تلقائيًّا. والمِرساة المُنزَّلة لا تُزال؛ بل تظل تُصادِق على أساس الجدارة كأي مُسجَّل.

الطرد بسبب التعطّل

المُصادِق الذي يبقى **صامتًا في فترته الزمنية** يُعامَل على أنه غير متصل و**يُطرَد من المجموعة النشطة عند حدود الحقبة التالية** (الحقب هي `epoch-height = 17000` من الكتل الرئيسية، ≈ 3 أيام بكتل كل 15 ثانية). أبقِ عقدتك قيد التشغيل ومُزامَنة وساعتها صحيحة كي لا تفوِّت فترة زمنية أبدًا.

المصادرة (slashing) بسبب التصويت المزدوج — شغِّل عقدة واحدة فقط لكل مفتاح

إنّ الإدلاء بصوتين متعارضين لنفس الفترة الزمنية/المرحلة هو **تصويت مزدوج** (equivocation). وهو قابل للإثبات: فالصوت الثاني المتعارض يُنتج معاملة `SlashingTx`، وتؤدّي `equivocation-ban-threshold = 2` من المخالفات المُثبَتة المتمايزة إلى **حظر دائم**. ويكسب المُبلِّغ مكافأة قدرها `slash-reporter-bounty = 5` من الخزينة.

العاقبة التشغيلية صارمة: **شغِّل عقدة واحدة لا غير لكل مفتاح توقيع**، و**أبقِ ساعة تلك العقدة مُزامَنة**. فعقدتان تتشاركان مفتاحًا، أو نسخة مُسيئة/مكرَّرة، قد تُنتجان أصواتًا متعارضة وتُسبِّبان حظر مفتاحك دائمًا. لا تنسخ `config.json` أبدًا إلى عقدة ثانية قيد التشغيل.

5. المراقبة والعمليات

الصحة والمقاييس (actuator)

تُقدَّم نقاط نهاية Spring Boot Actuator **مجرَّدةً** (ليس ضمن غلاف الواجهة البرمجية):

نقطة النهاية	الغرض
`actuator/health/`	فحص صحة الحيوية/الجاهزية.
`actuator/info/`	تفاصيل البناء/المعلومات.
`actuator/metrics/`	فهرس المقاييس وقيمها.
`actuator/prometheus/`	نقطة نهاية كَشْط Prometheus.

وجِّه Prometheus إلى `actuator/prometheus/` للمراقبة والتنبيه على أساس السلاسل الزمنية (وحدة المعالجة المركزية، والذاكرة، وJVM، وأعداد النظائر، وغيرها). يكشف مُشغِّل الحزمة العامة `actuator/health/` **فقط** افتراضيًّا (`--management.endpoints.web.exposure.include=health`)؛ أضِف `prometheus` إلى تلك القائمة في بيئة المُشغِّل لكَشْط عقدة مُنزَّلة، وأبقِ المُشغِّل (actuator) خلف جدار حماية أو وكيل عكسي. و`GET /api/v1/info/hardwareInfo` معطَّلة ما لم تُضبط `node.api.hardware-info=true`.

حالة السلسلة والعقدة (واجهة REST البرمجية)

نقطة النهاية	الغرض
`api/v1/explorer/info/`	نظرة عامة على الشبكة: `transactionsCount`، `blocksCount`، `nodesCount`، `circulatingSupply`، `maxSupply`، `validatorsCount`، `epochDate`، `epochNumber`، `transactionsPerSecond`، `blockProductionTime`.
`api/v1/info/`	هذه العقدة: `genesisHash`، `networkId`، `port`، `host`، `nodeKey`، `protocolVersion`، `apiVersion`، `version`، `chainSpecDigest`، `faucet`، `syncStatus`.
`api/v1/info/genesis/`	ملف `genesis.json` العام لهذه السلسلة (ما تُقلِع منه العقدة المُنزَّلة).
`api/v1/info/version/`	إصدار العقدة.
`api/v1/info/uptime/`	مدة تشغيل العقدة.
`api/v1/validators/active/`	المجموعة النشطة الحالية (مُقسَّمة على صفحات).

استخدم `api/v1/explorer/info/` للتأكد من تقدّم الكتل، و`api/v1/validators/active/` للتأكد من وجود مفتاحك في المجموعة النشطة.

السجلّات

- العقدة الأصلية/المفردة: المُخرَج القياسي (stdout) (وبيانات H2 ضمن دليل `/data` الخاص بالحزمة).
- شبكة العُقَد المتعددة الداخلية: `run/logs/node{1,2,3}.log`؛ تابِعها جميعًا بـ `run/nodes.sh logs/.` (انظر §6).

بوّابة SYNCHRONIZED

الإنتاج مشروط بحالة مزامنة العقدة. لا تُنتج العقدة وتُصوِّت إلّا بعد أن تبلغ حالة **SYNCHRONIZED** (وفقط ما دامت ساعتها مُزامَنة عبر NTP). وبعد إعادة التشغيل أو عند الانضمام إلى الشبكة، توقَّع أن تُزامِن العقدة أولًا — فهي تسير عبر الحقب رجوعًا لسدّ الفجوات، وتتحقق لكل حقبة، وتحفظ تقدّمًا قبل أن تستطيع الإنتاج. إذا كانت عقدتك قيد التشغيل لكنها لا تُنتج، فتحقّق من أنها مُزامَنة وساعتها مُزامَنة قبل أي شيء آخر.

النسخ الاحتياطي

خُذ نسخًا احتياطية، حسب ترتيب الأهمية:

1. **`data/config.json`** (`config.json` في التخطيط الداخلي) — مفتاح التوقيع وهوية العقدة. فقدان المفتاح قابل للاسترداد (يمكنك إعادة التسجيل ودفع الوديعة مجددًا)، لكن المفتاح **المُسرَّب** خطير؛ احمِه كأي مفتاح خاص.
2. **قاعدة البيانات** — ملفات H2 ضمن `/data` (أو قاعدة بيانات PostgreSQL الخاصة بك). يمكن إعادة بناء الحالة بإعادة المزامنة من النظائر، لكن النسخة الاحتياطية تُسرِّع الاسترداد.

واحتفظ أيضًا بـ **عبارة استرداد محفظتك (seed)** احتياطيًا دون اتصال — فهي تتحكّم بالمحفظة التي تتلقّى مكافآتك ودفعت الوديعة.

الترقيات

1. نزِّل الحزمة الجديدة لمنصّتك وتحقَّق من `SHA256SUMS`.
2. أوقِف العقدة.
3. فكّ ضغط الحزمة الجديدة فوق المجلد (أو اسحب صورة Docker الجديدة)؛ **أبقِ دليل `/data`** (الهوية + دفتر الأستاذ) في مكانه — وفي Docker هو وحدة التخزين المُركَّبة.
4. أعِد التشغيل وتأكَّد من أن العقدة تبلغ SYNCHRONIZED، وللمُصادِق، أنها تعود إلى المجموعة النشطة (`api/v1/validators/active/`).

رقِّ المُصادِقين واحدًا تلو الآخر كي تُبقي الشبكة على نِصاب حيّ — إلا إذا ذكرت ملاحظات الإصدار أن إصدار بروتوكول الشبكة قد تغيّر، وعندها يجب أن تنتقل كل العقد إلى الإصدار الجديد (فالإصداران القديم والجديد يرفض كلٌّ منهما إطارات الآخر).

6. كيف تُوزَّع العقدة

العقدة **مغلقة المصدر وتُوزَّع كملفات تنفيذية أصلية فقط** — لا يوجد ملف JAR ولا حاجة إلى تثبيت JDK ولا شيء يُبنى:

المنصّة	الأرشيف	التشغيل
Linux x86-64	`janzeer-node-<version>-<network>-linux-x64.tar.gz`	`start.sh/.` (مرفق معها وحدة systemd باسم `janzeer-node.service`)
لينكس arm64 (راسبيري باي 3/4/5، نظام 64 بت)	`janzeer-node-<version>-<network>-linux-arm64.tar.gz`	`start.sh/.`
Windows x86-64	`janzeer-node-<version>-<network>-windows-x64.zip`	`start.cmd`
macOS (Apple silicon / Intel)	`janzeer-node-<version>-<network>-macos-{arm64,x64}.tar.gz`	`start.sh/.`

تُنشر المنصّات تباعًا عند بنائها — وتُظهر صفحة **المُصادِقون** في janzeer.org غير المتاح منها بعبارة «قريبًا»، وتعرض بصمة SHA-256 لكل أرشيف إلى جانب `SHA256SUMS`. تحتوي الحزمة على الملف التنفيذي `janzeer` (وبداخله المستكشف ومحفظة الويب)، وملف

`genesis.json` العام للشبكة، والمُشغِّل، و`README.txt` — **بلا مفاتيح وبلا `config.json`**: تُنشئ عقدتك هويتها بنفسها عند أول تشغيل (§2، الخطوة 3).

حزمة واحدة = شبكة واحدة: معرّف الشبكة وبصمة كتلة التكوين مثبّتان في المُشغِّل، فلا يمكن لحزمة شبكة الاختبار أن تنضم إلى الشبكة الرئيسية. للترقية، فُكّ الحزمة الجديدة بجوار القديمة واحتفظ بمجلد `/data` (§5 *الترقيات*). تذكر ملاحظات الإصدار **إصدار بروتوكول P2P**؛ وعند تغيّره ترفض الشبكةُ عُقدَ الإصدار القديم حتى تُرقّى.

7. تصليب الإنتاج وقائمة مراجعة الأمان

- ☐ **أبدِل جميع أسرار التطوير/شبكة الاختبار.** إنّ أسرار المراسي المُلتزَم بها والقيم الافتراضية للكتلة التكوينية للتطوير/شبكة الاختبار فقط — أنشئ مفاتيح جديدة وكتلة تكوينية جديدة لأي نشر حقيقي.
- ☐ **صنبور صندوق التطوير مُعطّل.** تأكّد من أن `JANZEER_DEV_FUND` / `janzeer.dev-fund` غير مُعيَّن في الإنتاج.
- ☐ **عقدة واحدة لكل مفتاح.** لا تُشغِّل أبدًا عقدتين بنفس `secret`، ولا تنسخ أبدًا `config.json` إلى نسخة ثانية حيّة — يؤدّي التصويت المزدوج إلى حظر دائم.
- ☐ **مزامنة الساعة.** عيِّن `node.ntp-servers` موثوقة وراقِب انحراف (`ntp-offset-threshold = 250 ms` NTP)؛ فالعقدة ذات الساعة السيئة تتوقف عن الإنتاج بصمت.
- ☐ **لا تُعدِّل `*.consensus` أبدًا على شبكة حيّة.** هذه المعاملات مثبَّتة على بصمة مواصفات السلسلة عند الكتلة التكوينية ويُتحقَّق منها في المصافحة؛ وتغيير واحدة منها يجعل عقدتك تنشقّ عن الشبكة. لا تمسّ سوى الإعدادات التشغيلية (المنافذ، وقاعدة البيانات، والنظائر، وNTP).
- ☐ **حافظ على نِصاب حيّ.** تحتاج النهائية إلى `floor(2N/3) + 1` من المجموعة النشطة؛ أبقِ عددًا كافيًا من المُصادِقين قيد التشغيل (المراسي الثلاث جميعها للشبكة الافتراضية، و≤ 4 مُصادِقين لتحمّل عطلٍ واحد) ورقٍّ واحدًا تلو الآخر.
- ☐ **احمِ المفاتيح.** قيِّد أذونات الملفات على `config.json` (و`run/secrets.env` لإعداد العُقَد المتعددة الداخلي)؛ أبقِها خارج التحكّم بالإصدار وخارج السجلّات/النسخ الاحتياطية التي يمكن لآخرين قراءتها. خُذ نسخة احتياطية من عبارة استرداد محفظتك دون اتصال.
- ☐ **ضع جدار حماية على المنافذ.** اكشف P2P (`9199`، أو `9190/9191/9192`) فقط بقدر الحاجة للاتصال بالنظائر؛ قيِّد أو وكِّل واجهة REST البرمجية (`7019`) ولا سيما نقاط نهاية `**/actuator/` — لا تكشف actuator علنًا.
- ☐ **استخدم PostgreSQL مُدارة** للإنتاج (`DB_URL/DB_USERNAME/DB_PASSWORD`) بنسخها الاحتياطية الخاصة، بدلًا من H2 المُضمَّنة الافتراضية.
- ☐ **تخزين مستمر.** أبقِ دليل `/data` الخاص بالحزمة و`config.json` على تخزين دائم (بالنسبة إلى Docker، رَكِّب وحدة تخزين) كي تنجو الهوية والحالة من إعادات التشغيل.
- ☐ **راقِب ونبِّه.** اكشط `actuator/prometheus/`، وراقِب `api/v1/explorer/info/` (تقدّم السلسلة) و`api/v1/validators/active/` (وجود مفتاحك)، ونبِّه على التعطّل كي لا تُطرَد بسبب تفويت الفترات الزمنية.
- ☐ **خُذ نسخًا احتياطية بانتظام.** احتفظ بنسخ احتياطية آمنة وحديثة من `config.json` وقاعدة البيانات وعبارة استرداد محفظتك و(للكتلة التكوينية المخصَّصة) بذور المراسي.

حالة الوثيقة — الإصدار 1.2 (حزمة العقدة العامة، genesis.json؛ الشبكة الرئيسية هي الشبكة العامة الوحيدة، ولا صنبور عام)؛ يعكس إصدار العقدة 0.1.14 / بروتوكول الاتصال 3.4.0. تعيش الحقائق الدقيقة على مستوى البايت (التعمية، وتخطيطات المعاملات/الكتل، وحسابات الإجماع) في الورقة الصفراء.