

ICE, STUN, TURN — production cheat sheet

Companion to the 2026 deep-dive article. Tape it next to the monitor.

1. Candidate priority — RFC 8445 §5.1.2

$$\text{priority} = (2^{24}) \times \text{type_preference} + (2^8) \times \text{local_preference} + (256 - \text{component_id})$$

Recommended type_preference: host 126 · peer-reflexive 110 · server-reflexive 100 · relay 0

Type	From	Gather time	When it wins	Priority
Host (126)	OS-reported address on a real interface	< 5 ms	Same-LAN peers; mesh on local network	2 130 706 431
Peer-reflexive (110)	Discovered during connectivity check itself	during check	Symmetric NAT mappings created mid-call	1 862 270 975
Server-reflexive (100)	STUN binding response (XOR-MAPPED-ADDRESS)	20-200 ms	Cone-NAT-to-cone-NAT direct paths	1 694 498 815
Relay (0)	TURN Allocate response (RFC 8656)	100-500 ms	Symmetric NAT, UDP-blocked, AI agents (100%)	16 776 960

2. iceConnectionState — what to do in each state

State	Caused by	Recommended application action
new	PC just created	no-op; gathering has not started yet
checking	gathering complete, checks running	show 'connecting...' spinner; expect < 1 s here
connected	first pair valid; media flowing	clear spinner; cancel any pending restart timer
completed	every pair in checklist tested	treat identically to 'connected' in most apps
disconnected	consent freshness probes failing	wait 5 s; ≈ 80% recover; then call restartIce()
failed	every pair has failed	restartIce() is mandatory; the state cannot self-recover
closed	pc.close() called	terminal; create a new RTCPeerConnection if needed

3. iceServers configuration template — 2026

```
new RTCPeerConnection({
  iceServers: [
    { urls: 'stun:stun.l.google.com:19302' },
    { urls: [
      'turn:turn.example.com:3478?transport=udp', // cheap path
      'turn:turn.example.com:3478?transport=tcp', // for UDP-blocked nets
      'turn:turn.example.com:443?transport=tcp' // for everything-blocked nets
    ] },
    { username: shortLivedUsername, // HMAC-SHA1, server-issued, 600 s TTL
      credential: shortLivedPassword,
      iceTransportPolicy: 'all', // 'relay' = force every byte through TURN
      bundlePolicy: 'max-bundle', // one transport for all tracks (RFC 9143)
      tcpMuxPolicy: 'require' // RTP + RTCP on one port
    }
  ]
});
```

4. TURN deployment — pick the right pattern

Pattern	Time to ship	Cost @ 100 TB	Cost @ 10 PB	On-call	Fit
Managed (Cloudflare, Twilio)	1 day	\$5,000	\$500,000	none	< 10K MAU, no infra team
Self-hosted coturn / eternal	1 week	\$1,000	\$25,000	high	mature infra team
Pion / STUNner on K8s	1-2 weeks	\$1,000	\$25,000	medium	Kubernetes-native shops
Hybrid (self + managed fallback)	1-2 weeks	\$1,000	\$25,000+	low	belt-and-braces production

5. Eight pitfalls — verify before launch

- STUN-only ICE servers (always include TURN)
- Restarting ICE on every transient disconnect
- TURN over UDP only (also expose TCP/3478 and TLS/443)
- Single-region TURN (deploy 3+ regions for global)
- Long-lived TURN credentials in client JavaScript
- Not budgeting for the 100% relay case (AI agents)
- Ignoring iceconnectionstatechange events
- Trusting iceTransportPolicy:'relay' as anonymity