

Premarket Momentum Bot — Daily Operation Flow

monitor.py's SessionOrchestrator, one run per trading day — generated 2026-09-01

1. STARTUP (cron launches the process any time before the scan)

config_loader loads + validates config.json — fails fast, listing every problem, if any lookback / percentage / score field is zero or negative • acquires the singleton file lock state/monitor.pid.lock (a 2nd instance refuses to start against the same account) • prunes prior-day entries from positions.json and intraday_health.json • constructs PositionManager + StreamManager

2. 09:29 ET — ONE FULL-UNIVERSE SCAN (premarket_scanner.py)

Alpaca tradable assets ≈ 13,400
↓ symbol_filters.py drops ETF/ETN/trust, leveraged-inverse, OTC, overly-complex names
≈ 3,400 survive asset filtering
↓ prefilter_by_snapshot(): keep only 5–15 price band + a minimum prior-day-volume floor
≈ 550 survive the price/volume prefilter
↓ scorer.py scores each survivor — 13 weighted factors (RVOL, momentum, VWAP structure, consolidation, distance from PM high, spread, extension-risk penalty ...) → total_score 0–100
Top candidates.premarket_candidate_count (30) kept → premarket_20
final_10 = premarket_20's own top slice by total_score → watchlist.json + top_stocks.json written

3. START SIP STREAM (stream.py)

Subscribes trades / quotes / 1-min bars for every premarket_20 symbol • each symbol gets its own rolling in-memory SymbolBuffer • runs on a background thread with reconnect + backoff, so monitor.py's main loop stays simple synchronous polling against the buffer, never the network directly

4. WAIT FOR MARKET OPEN — 09:30 ET

5. MAIN TRADING LOOP — every 5s (schedule.poll_interval_seconds_intraday), 09:30 → 15:55

5a. Poll pending exits

Advances any close order already in flight (avoids resubmitting a second close while the first is still settling at the broker).

5c. Scan for entries

Rank premarket_20 by cached health_score → shortlist top 5 (entry_shortlist_size), skipping any benched after repeated failures. Each gets a FRESH health check, then entry_engine.evaluate_entry(): breakout of resistance (pm_high) held • VWAP above & rising • volume expansion • higher-lows structure • momentum not fading • RSI<70 • spread/extension limits • fresh health ≥ floor → risk_manager sizes the order and position_manager buys.

5e. [every 60s] Update intraday health

Every premarket_20 symbol re-scored on price-slope / VWAP-support / volume / structure → HEALTHY→WATCH→STALE→UNHEALTHY→REMOVED, with hysteresis so one noisy read can't flip the state.

5b. Manage open positions

risk_manager updates the trailing stop (ATR × 0.25, only ever moves up). If touched: intraday_health checks for a CONFIRMED fade before exiting — unconfirmed dips get one bounded grace widening first.

5d. Reconcile with broker

Compares local ledger vs Alpaca's actual open positions; logs mismatches only — never auto-corrects.

5f. [every 30 min] Full intraday rescan

Re-scores the ENTIRE ≈550-symbol prefiltered universe (not just the current 20) and rotates premarket_20 to the fresh top N — this is how a stock that missed the 09:29 cut can still be discovered later.

Shared state (data_store.py)

watchlist.json

positions.json

intraday_health.json

top_stocks.json

cooldowns.json

— File-based hand-off: every module reads/writes through here instead of calling each other directly.

6. ON POSITION EXIT → top_stocks.find_replacement()

Refreshes intraday_health for the WHOLE premarket_20 pool (not just eligible fillers) so no symbol's state goes stale between 60s cycles • ranks symbols that aren't held/on-cooldown and are health-eligible by health_score • must clear trading.min_replacement_health_score or the slot is left empty this cycle → winner still has to pass entry_engine.evaluate_entry() before monitor.py actually buys it.

7. END OF DAY

15:15 no_new_entries_after — _scan_for_entries() stops firing, open positions keep trailing
15:55 force_liquidate_time — liquidate_all() closes every open/closing position, retried for up to eod_liquidation_max_wait_seconds (120s) so same-day fills aren't pushed into tomorrow's log
16:00 market_close_time — today's trades/wins/losses/P&L summary written, SIP stream stopped, exit

repeats every 5s

reads / writes