JARVIS + THE TOOL — Packaging Kit v0.1

Goal: Package a Windows-first automation stack so we can charge now. This kit includes the repo layout, installer scripts, CLI, template schema, and ready-to-use marketing blocks.

---

0) Bootstrap in one go (PowerShell)

Run this once in the folder where you want the product. It generates the full tree and files.

```
# ===== JARVIS Packaging Bootstrap v0.1 =====
$ROOT = Join-Path (Get-Location) "jarvis-pack-v0_1"
$dirs = @(
  'core/operator-bridge', 'core/botkit', 'cli', 'templates/deploy-vercel',
  'templates/metrics-digest', 'templates/lead-magnet', 'packaging',
  'console', 'marketing/landing', 'marketing/copy'
)
$dirs | ForEach-Object { New-Item -ItemType Directory -Force -Path (Join-Path
$ROOT $_) | Out-Null }

# --- CLI ---
@'
param(
  [Parameter(Position=0)]
[ValidateSet('init','up','down','status','logs','run','template')] $cmd =
'status',
  [Parameter(Position=1)] $arg
)
$ErrorActionPreference = 'Stop'
$Base = Split-Path -Parent $MyInvocation.MyCommand.Path
$Root = Resolve-Path (Join-Path $Base '..')
$Node = (Get-Command node -ErrorAction SilentlyContinue)?.Source
if (-not $Node) { Write-Error 'Node.js not found. Install LTS from nodejs.org';
exit 1 }

function Invoke-Bridge { param([string]$Fn,[string]$A)
  Import-Module (Join-Path $Root 'core/operator-bridge/bridge.psm1') -Force
  & $Fn $A
}

switch ($cmd) {
  'init' {
    Copy-Item (Join-Path $Root 'packaging/.env.example') (Join-Path $Root
'.env') -Force
```

```
    Write-Host 'Initialized .env; edit it before running.'
  }
  'up' {
    Invoke-Bridge -Fn Start-Jarvis -A $null
  }
  'down' {
    Invoke-Bridge -Fn Stop-Jarvis -A $null
  }
  'status' {
    Invoke-Bridge -Fn Get-JarvisStatus -A $null
  }
  'logs' {
    Get-Content (Join-Path $Root 'core/botkit/jarvis.log') -Wait -ErrorAction
SilentlyContinue
  }
  'run' {
    Push-Location $Root
    node .\core\botkit\server.js
    Pop-Location
  }
  'template' {
    if (-not $arg) { Write-Error 'Usage: .\\jarvis.ps1 template <template-id>';
exit 1 }
    Invoke-Bridge -Fn Invoke-JarvisTemplate -A $arg
  }
}
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'cli/jarvis.ps1')

# --- Operator Bridge (PowerShell) ---
@'
# core/operator-bridge/bridge.psm1
$ErrorActionPreference = 'Stop'
$Root = Split-Path -Parent $PSScriptRoot
$Root = Split-Path -Parent $Root  # project root
$TaskName = 'JarvisAgent'

function Start-Jarvis {
  $node = (Get-Command node -ErrorAction SilentlyContinue)?.Source
  if (-not $node) { throw 'Node.js LTS is required' }
  $wd  = $Root
  $exe = $node
  $arg = 'core\\botkit\\server.js'
  $act = New-ScheduledTaskAction -Execute $exe -Argument $arg -WorkingDirectory
$wd
  $trg = New-ScheduledTaskTrigger -AtLogOn
  $set = New-ScheduledTaskSettingsSet -RestartCount 3 -RestartInterval (New-
TimeSpan -Minutes 1) -ExecutionTimeLimit (New-TimeSpan -Hours 0)
  Register-ScheduledTask -TaskName $TaskName -Action $act -Trigger $trg -
```

```
Description 'JARVIS BotKit' -Settings $set -Force | Out-Null
  Start-ScheduledTask -TaskName $TaskName
  "Started $TaskName"
}

function Stop-Jarvis { if (Get-ScheduledTask -TaskName $TaskName -ErrorAction
SilentlyContinue) { Unregister-ScheduledTask -TaskName $TaskName -Confirm:
$false } "Stopped $TaskName" }
function Get-JarvisStatus { Get-ScheduledTask -TaskName $TaskName -ErrorAction
SilentlyContinue | Select-Object TaskName,State,LastRunTime,LastTaskResult }
function Invoke-JarvisTemplate([string]$Id) {
  $tpl = Join-Path $Root ("templates/$Id/run.ps1")
  if (-not (Test-Path $tpl)) { throw "Template not found: $Id" }
  & $tpl
}
Export-ModuleMember -Function *
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'core/operator-bridge/
bridge.psm1')

# --- BotKit (Node) ---
@'
{
  "name": "jarvis-botkit",
  "private": true,
  "type": "module",
  "version": "0.1.0",
  "dependencies": {
    "express": "^4.19.2",
    "dotenv": "^16.4.5"
  },
  "scripts": { "start": "node server.js" }
}
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'core/botkit/package.json')

@'
import 'dotenv/config'
import fs from 'node:fs'
import path from 'node:path'
import express from 'express'

const app = express()
app.use(express.json())
const PORT = process.env.PORT || 4310
const LOG  = path.join(process.cwd(), 'core', 'botkit', 'jarvis.log')

function log(...a){ const line = `[${new Date().toISOString()}] `+a.join(' ')
+"\n"; fs.appendFileSync(LOG, line) }
```

```
app.get('/health', (_,res)=>res.json({ok:true, t:Date.now()}))
app.post('/run', async (req,res)=>{
  const { template } = req.body||{}
  if(!template) return res.status(400).json({ok:false, error:'template
required'})
  log('RUN', template)
  res.json({ok:true, template})
})

app.listen(PORT, ()=>{
  log('Jarvis BotKit up on', PORT)
  console.log('Jarvis BotKit on http://localhost:'+PORT)
})
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'core/botkit/server.js')

# --- Templates ---
@'
{
  "id": "deploy-vercel",
  "name": "Static Site Autodeploy (Vercel)",
  "version": "1.0.0",
  "inputs": [
    {"key":"repo_url","label":"Git repo URL","required":true},
    {"key":"vercel_token","label":"Vercel Token","required":true}
  ],
  "steps": ["pull","build","deploy"],
  "success": "HTTP 200 from deployment endpoint"
}
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'templates/deploy-vercel/
template.json')

@'
param()
Write-Host 'Deploying to Vercel...'
# Placeholder: add your vercel CLI call here
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'templates/deploy-vercel/
run.ps1')

@'
{
  "id": "metrics-digest",
  "name": "Daily Metrics Digest",
  "version": "1.0.0",
  "inputs": [
    {"key":"sheet_id","label":"Google Sheet ID","required":true},
    {"key":"send_to","label":"Email/WA","required":true}
  ],
  "steps": ["fetch","format","send"],
```

```
  "success": "+ delivered digest"
}
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'templates/metrics-digest/
template.json')

@'
param()
Write-Host 'Sending daily metrics digest...'
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'templates/metrics-digest/
run.ps1')

@'
{
  "id": "lead-magnet",
  "name": "Lead Magnet + Email Drip",
  "version": "1.0.0",
  "inputs": [
    {"key":"asset_url","label":"Download URL","required":true},
    {"key":"email_api_key","label":"Resend/Mailgun Key","required":true}
  ],
  "steps": ["capture","deliver","drip"],
  "success": "+ subscribers and open rate"
}
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'templates/lead-magnet/
template.json')

@'
param()
Write-Host 'Delivering lead magnet and starting drip...'
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'templates/lead-magnet/
run.ps1')

# --- Packaging (.env + installer) ---
@'
# .env example
PORT=4310
# add provider tokens here when needed
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'packaging/.env.example')

@'
param()
$ErrorActionPreference = 'Stop'
$Base = Split-Path -Parent $MyInvocation.MyCommand.Path
$Root = Split-Path -Parent $Base

if (-not (Test-Path (Join-Path $Root '.env'))) {
  Copy-Item (Join-Path $Base '.env.example') (Join-Path $Root '.env') -Force
}
```

```
# install scheduled task
Import-Module (Join-Path $Root 'core/operator-bridge/bridge.psm1') -Force
Start-Jarvis
Write-Host 'Installed and started JARVIS. Use .\\cli\\jarvis.ps1 status'
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'packaging/install.ps1')

@'
param()
Import-Module (Join-Path (Split-Path -Parent $PSScriptRoot) 'core/operator-
bridge/bridge.psm1') -Force
Stop-Jarvis
Write-Host 'JARVIS uninstalled.'
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'packaging/uninstall.ps1')

# --- Console placeholder ---
@'
# THE TOOL (Ops Console)
Initial version links to http://localhost:4310/health for status. GUI ships in
v0.2 (Tauri).
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'console/README.md')

# --- Marketing blocks ---
@'
<section class="pricing">
  <h2>Pricing</h2>
  <div class="grid">
    <div class="card"><h3>Starter</h3><p>$29/mo</p><ul><li>1 project</li><li>3
templates</li></ul></div>
    <div class="card"><h3>Pro</h3><p>$99/mo</p><ul><li>5 projects</li><li>All
templates</li><li>Webhook + Schedules</li></ul></div>
    <div class="card"><h3>Business</h3><p>$249/mo</p><ul><li>15 projects</
li><li>Team seats</li><li>Priority fixes</li></ul></div>
    <div class="card"><h3>Agency</h3><p>$499/mo</p><ul><li>Unlimited projects</
li><li>White-label console</li><li>SLA</li></ul></div>
  </div>
</section>
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'marketing/landing/
pricing.html')

@'
# Value Proposition (site copy)
Operator OS for Windows-first teams. Launch automations that make money fast.
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'marketing/copy/value-prop.md')

@'
# Affiliate Program (one-pager)
```

```
- 30% recurring for 12 months
- Payout monthly via Stripe
- Simple link: https://aogrl.com/aff/yourcode
'@ | Set-Content -Encoding utf8 (Join-Path $ROOT 'marketing/copy/affiliates.md')

Write-Host "\nPack created at: $ROOT"
Write-Host "Next: 1) edit $ROOT\\.env  2) .\\packaging\\install.ps1  3) .\\cli\
\jarvis.ps1 status"
```

---

1) Repo layout

```
jarvis-pack-v0_1/
  cli/                    # jarvis.ps1 (init/up/down/status/logs/template)
  core/
    operator-bridge/      # PowerShell module to install/uninstall scheduled
task
    botkit/               # Node server (health + /run)
  templates/
    deploy-vercel/        # template.json + run.ps1
    metrics-digest/
    lead-magnet/
  packaging/              # .env.example, install.ps1, uninstall.ps1
  console/                # THE TOOL placeholder
  marketing/              # pricing HTML + copy
```

2) SKU & Licensing (ready to ship)

- **Starter $29**: 1 project, 3 templates, local.
- **Pro $99**: 5 projects, all templates, webhook + schedules.
- **Business $249**: 15 projects, team seats, priority fixes.
- **Agency $499**: unlimited projects, white-label console, SLA.
- **License**: BUSL-1.1 for core; commercial EULA for templates/add-ons. Trial 7 days, cancel anytime.

3) How to package for delivery

- Zip `jarvis-pack-v0_1` + checksum.
- Add `install.ps1` instructions.
- Stripe checkout → Customer portal → License email with download link.

4) Roadmap (fast)

- v0.2: Console GUI (Tauri), template inputs UI, logs.
- v0.3: Template marketplace + auto-updates.

- v0.4: Windows service wrapper (optional), Linux support.

5) GTM snippets included

- Pricing HTML block.
- Value-prop copy.
- Affiliate one-pager.

Decision rule: If < 10 paying users or < 3 agencies within 30 days, narrow to 2 money templates and push Agency-only.