Sample Scripts & Automation Recipes

This section provides practical, real-world examples showing how to interact with the AiVRIC API for automation, reporting, and integration.
All examples assume:

- You have a valid **API token**
- You understand the basics of **AiVRIC API authentication & rate limits**
- Your environment allows outbound HTTPS to the AiVRIC API endpoint

Note: These are *recipes*, not full applications—each example is intentionally short, composable, and production-ready.

---

1. Fetch Latest Posture Snapshot (Python)

Retrieve the latest cloud posture assessment for your organization.

```
import requests

API_KEY = "YOUR_AIVRIC_API_KEY"
BASE_URL = "https://api.aivric.io/v1"

headers = {
  "Authorization": f"Bearer {API_KEY}"
}

resp = requests.get(f"{BASE_URL}/posture/latest", headers=headers)

if resp.status_code == 200:
  snapshot = resp.json()
  print("Posture Score:", snapshot["score"])
else:
  print("Error:", resp.text)
```

Use case:
Automate nightly posture checks and pipe results into Slack or a dashboard.

---

2. Pull All Open High-Severity Findings

```
import requests

API_KEY = "YOUR_AIVRIC_API_KEY"
BASE_URL = "https://api.aivric.io/v1"

params = {
  "severity": "high",
  "status": "open"
}

resp = requests.get(
  f"{BASE_URL}/findings",
  headers={"Authorization": f"Bearer {API_KEY}"},
  params=params
)

data = resp.json()

for f in data["results"]:
  print(f"{f['id']} - {f['title']} - {f['resource']}")
```

Use case:
Sync high-severity findings into **SIEM**, **Jira**, or **ServiceNow**.

---

3. Auto-Create Jira Tickets for New Findings

Minimal example using Python + Jira Cloud API.

```
import requests

AIVRIC_API_KEY = "YOUR_AIVRIC_KEY"
JIRA_API_TOKEN = "YOUR_JIRA_API_TOKEN"
JIRA_EMAIL = "you@example.com"
JIRA_URL = "https://your-domain.atlassian.net"
JIRA_PROJECT_KEY = "SEC"

# Step 1: Get new findings
findings = requests.get(
  "https://api.aivric.io/v1/findings?status=open&new=true",
  headers={"Authorization": f"Bearer {AIVRIC_API_KEY}"}
).json()["results"]

# Step 2: Create Jira issues
for f in findings:
```

```
  payload = {
    "fields": {
      "project": {"key": JIRA_PROJECT_KEY},
      "summary": f"[AiVRIC] {f['title']}",
      "description": f"Resource: {f['resource']}\n\nDetails:\n{f['description']}",
      "issuetype": {"name": "Task"}
    }
  }

  resp = requests.post(
    f"{JIRA_URL}/rest/api/3/issue",
    json=payload,
    auth=(JIRA_EMAIL, JIRA_API_TOKEN)
  )

  print("Ticket created:", resp.text)
```

Use case:
Fully automated compliance backlog creation managed via your existing ITSM workflow.

---

4. Export Evidence Bundle for PCI/SOC2

```
curl -X GET "https://api.aivric.io/v1/evidence/export?framework=SOC2" \
   -H "Authorization: Bearer $AIVRIC_API_KEY" \
   -o evidence_soc2.zip
```

Use case:
Daily or weekly automated evidence generation for auditors, QSAs, or internal GRC teams.

---

5. Trigger an On-Demand Scan (AWS Example)

```
curl -X POST "https://api.aivric.io/v1/scans" \
 -H "Authorization: Bearer $AIVRIC_API_KEY" \
 -H "Content-Type: application/json" \
 -d '{
     "provider": "aws",
     "account_id": "123456789012",
     "scan_type": "full"
   }'
```

Use case:
Run posture scans before major releases, change windows, or compliance deadlines.

6. Send Alerts to Slack

```
import requests

AIVRIC_API_KEY = "YOUR_AIVRIC_KEY"
SLACK_WEBHOOK = "YOUR_SLACK_WEBHOOK"

latest = requests.get(
  "https://api.aivric.io/v1/alerts/latest",
  headers={"Authorization": f"Bearer {AIVRIC_API_KEY}"}
).json()

message = {
  "text": f"🔥 *New AiVRIC Alert*\n{latest['title']}\nSeverity: {latest['severity']}"
}

requests.post(SLACK_WEBHOOK, json=message)
```

Use case:
Instant visibility for security or cloud teams.

---

7. Bulk Acknowledge Resolved Findings

Automatically clean up findings that AiVRIC confirms as fixed.

```
import requests

AIVRIC_API_KEY = "YOUR_AIVRIC_KEY"

# Step 1: Get resolved findings
resolved = requests.get(
  "https://api.aivric.io/v1/findings?status=resolvable",
  headers={"Authorization": f"Bearer {AIVRIC_API_KEY}"}
).json()

# Step 2: Bulk acknowledge
ids = [f["id"] for f in resolved["results"]]

resp = requests.post(
  "https://api.aivric.io/v1/findings/acknowledge",
  json={"ids": ids},
  headers={"Authorization": f"Bearer {AIVRIC_API_KEY}"}
)

print("Acknowledged:", resp.json())
```

Use case:
Keeps dashboards clean + supports automated devsecops pipelines.

---

8. Example GitHub Action CI Workflow

Runs a scan → uploads results → fails pipeline if critical findings exist.

```
name: AiVRIC Security Scan

on: [push]

jobs:
  aivric-scan:
    runs-on: ubuntu-latest
    steps:
      - name: Run AiVRIC CLI
        run: |
          ./aivric.exe scan --api-key ${{ secrets.AIVRIC_API_KEY }} --out results.json

      - name: Upload results
        run: |
          curl -X POST "https://api.aivric.io/v1/results/upload" \
            -H "Authorization: Bearer ${{ secrets.AIVRIC_API_KEY }}" \
            -F "file=@results.json"

      - name: Fail if critical issues
        run: |
          jq '.findings[] | select(.severity=="critical")' results.json && exit 1 || exit 0
```

Use case:
Shift-left enforcement in CI/CD.