Below is a **clean, production- grade HubSpot–only example** that uses **Zapier Webhooks** both **as a Trigger** *and* **as an Action**, and is **different from all prior prompts**.

The flow is **HubSpot → Zapier → HubSpot**, using **different HubSpot APIs on each side**.

✅HubSpot Webhooks – Real- World Zapier Implementation

Trigger + Action (HubSpot CRM Events → HubSpot Tickets API)

🧩 Business Scenario

When a **HubSpot Contact's lifecycle stage changes to "Customer"**, automatically **create a Support Ticket** in HubSpot so Customer Success can begin onboarding.

This pattern is common in:

- SaaS onboarding
- Handoff from Sales → Support
- Enterprise CRM automation

1️⃣ Webhook as Trigger

HubSpot → Zapier (contact.propertyChange)

HubSpot sends webhooks for CRM changes such as:

- contact.creation
- contact.propertyChange
- deal.propertyChange

We'll use **contact.propertyChange** on **lifecycle stage**.

A. Create the Zap Trigger (Zapier)

1. Create a new Zap
2. Trigger App: **Webhooks by Zapier**
3. Trigger Event: **Catch Hook**
4. Zapier generates a webhook URL:

```
https://hooks.zapier.com/hooks/catch/123999/hs-contact
```

Zapier's Catch Hook accepts POST requests and parses JSON automatically [1]

B. Configure HubSpot Webhook Subscription

In a **HubSpot Developer App** → Webhooks:

Subscription

```
{
  "subscriptionType": "contact.propertyChange",
  "propertyName": "lifecyclestage",
  "active": true
}
```

Target URL

```
https://hooks.zapier.com/hooks/catch/123999/hs-contact
```

HubSpot webhooks push updates in near real- time without polling [2]

C. Payload Sent by HubSpot

HubSpot batches events and sends an array:

```
[
  {
    "eventId": 1004321,
    "subscriptionType": "contact.propertyChange",
    "objectId": 567890,
    "propertyName": "lifecyclestage",
    "propertyValue": "customer",
    "portalId": 98765,
    "occurredAt": 1735614400000
  }
]
```

Each entry includes the **objectId**, property name, and new value [3]

---

[1] https://help.zapier.com/hc/en-us/articles/8496288690317-Trigger-Zaps-from-webhooks

[2] https://developers.hubspot.com/docs/api-reference/webhooks-webhooks-v3/guide

[3] https://inventivehq.com/blog/hubspot-webhooks-guide

D. Filter in Zapier

Add a **Filter** step:

```
Only continue if propertyValue equals "customer"
```

✅Trigger portion complete

2□□ Webhook as Action

Zapier → HubSpot Tickets API

Now Zapier will **call back into HubSpot** using a **different API** to create a Ticket.

A. Action Setup (Zapier)

- Action App: **Webhooks by Zapier**
- Event: **POST**
- URL:

```
https://api.hubapi.com/crm/v3/objects/tickets
```

Zapier can send authenticated POST requests to any REST API [4]

B. Headers

```
Authorization: Bearer {{HUBSPOT_PRIVATE_APP_TOKEN}}
Content-Type: application/json
```

HubSpot supports OAuth and Private App Bearer tokens for CRM APIs [5]

C. Request Body

```
{
  "properties": {
    "subject": "New Customer Onboarding",
    "hs_pipeline": "0",
    "hs_pipeline_stage": "1",
```

---

[4] https://help.zapier.com/hc/en-us/articles/8496083355661-How-to-get-started-with-Webhooks-by-Zapier

[5] https://developers.hubspot.com/docs/api-reference/crm-tickets-v3/guide

```
    "hs_ticket_priority": "HIGH"
  },
  "associations": [
    {
      "to": {
        "id": "{{objectId}}"
      },
      "types": [
        {
          "associationCategory": "HUBSPOT_DEFINED",
          "associationTypeId": 16
        }
      ]
    }
  ]
}
```

- associationTypeId: 16 links the ticket to the Contact
- Tickets require pipeline and stage IDs, not labels [6]

D. HubSpot API Response

```
{
  "id": "34567123",
  "properties": {
    "subject": "New Customer Onboarding",
    "hs_ticket_priority": "HIGH"
  }
}
```

✅Ticket created automatically

🔁 End- to- End Architecture

```
Contact lifecycle updates in HubSpot
        ↓
HubSpot Webhook (contact.propertyChange)
        ↓
Zapier Catch Hook (Trigger)
```

[6] https://developers.hubspot.com/docs/api-reference/crm-tickets-v3/guide

```
          ↓
  Zapier Filter (customer only)
          ↓
  Zapier POST → HubSpot Tickets API
           ↓
   Onboarding ticket created
```

🔐 Security Notes

- HubSpot signs webhook payloads with X-HubSpot-Signature
- Signature validation can be added using Code- by- Zapier if required [7]
- Endpoints must respond in <5 seconds to avoid retries [8]

✅Why This Is a True Real- World Pattern

- Uses **different HubSpot APIs** on each side
- No polling, no cron jobs
- Automates internal CRM workflow
- Scales to thousands of contacts
- Common in SaaS & Service Hub setups

🧠 Optional Enhancements

- Lookup Contact email before ticket creation
- Route tickets based on plan or region
- Add onboarding notes via Engagements API
- Fan- out to Slack or PagerDuty

---

[7] https://developers.hubspot.com/docs/apps/legacy-apps/authentication/validating-requests

[8] https://developers.hubspot.com/docs/api-reference/webhooks-webhooks-v3/guide