# Splunk for DevOps

**Introduction to Splunk**

Splunk is a robust software platform designed for searching, monitoring, and analyzing large volumes of machine-generated data through a web interface. It is especially useful for managing logs and events, providing real-time operational intelligence. Splunk is capable of ingesting data from diverse sources, including applications, servers, network devices, and sensors. It's commonly utilized in IT operations, security, and business analytics to derive actionable insights and ensure system reliability.

**Splunk in DevOps**

In DevOps, Splunk plays a key role in monitoring, logging, and data analysis, enabling teams to track system performance, examine application logs, and analyze other operational data. This helps DevOps teams troubleshoot issues, enhance efficiency, and ensure security across their development and production environments.

---

**Installing Splunk on Linux (for DevOps Use)**

🔧 **Step-by-Step: Install Splunk on Linux**

✅ **Prerequisites:**

- 64-bit Linux (Ubuntu, CentOS, RHEL, Debian, etc.)
- Root or sudo access
- At least 2GB RAM (4GB+ recommended)
- Python not needed (Splunk is self-contained)

📦 **1. Download Splunk** Download the .tgz version for a CLI-based installation:

```
wget -O splunk.tgz
"https://download.splunk.com/products/splunk/releases/9.2.0/linux/splunk-9.2.0-xx
xxxxx-Linux-x86_64.tgz"
```

(Replace xxxxxxx with the actual build hash from Splunk Downloads).

📁 **2. Extract and Move to /opt**

```
tar -xvzf splunk.tgz
sudo mv splunk /opt/splunk
```

👤 **3. Create Splunk User (Optional)**

```
sudo useradd splunk
sudo chown -R splunk:splunk /opt/splunk
```

🚀 **4. Start Splunk** Switch to the Splunk directory, accept the license, and set the admin username and password:

```
cd /opt/splunk/bin
sudo ./splunk start --accept-license
```

🔁 **5. Enable Splunk at Boot**

```
sudo /opt/splunk/bin/splunk enable boot-start
```

**For non-root users:**

```
sudo /opt/splunk/bin/splunk enable boot-start -user splunk
```

🌐 **6. Access Splunk Web UI** Open a browser and go to:

http://<your-server-ip>:8000

Login using the admin credentials you set.

🔍 **7. Add Data Sources (Log Files)** In the web UI, navigate to: **Settings → Add Data → Monitor → Files & Directories** Select paths like /var/log, /var/log/syslog, /var/log/nginx, etc.

---

## 🧼 Uninstall Splunk (if needed)

**To uninstall Splunk, run the following commands:**

```
sudo /opt/splunk/bin/splunk stop
sudo rm -rf /opt/splunk
sudo rm -rf /opt/splunkforwarder
```

---

**Key Components of Splunk:**

Splunk Indexer

Purpose: It stores and processes the incoming data.

How it works: When data is received (from logs, metrics, etc.), the indexer:

Parses the data

Indexes it for fast search

Stores it in indexes

Why it's important: This is where the core processing happens so that data is searchable and retrievable quickly.

Splunk Search Head

Purpose: It allows users to search, analyze, and visualize data.

How it works: You can enter search queries here using the Splunk Search Processing Language (SPL).

Features:

Dashboards

Alerts

Reports

Why it's important: It gives users the interface to interact with data and gain insights.

Splunk Forwarder

Purpose: It sends raw data to the indexer.

Types:

Universal Forwarder (UF): Lightweight, only forwards raw data.

Heavy Forwarder (HF): Can parse and filter data before sending.

Why it's important: It’s used to collect data from various sources (servers, apps) and push it to the indexer.

Splunk Deployment Server

Purpose: It manages and deploys configurations (like apps, inputs, and outputs) to multiple Splunk forwarders.

Use Case: Centralized management of many forwarders in large environments.

Why it's important: Saves time and effort by automating configuration distribution.

---

## 🔄 Universal Forwarder

The Splunk Universal Forwarder (UF) is a lightweight agent that forwards logs to a Splunk Indexer.

**Universal Forwarder Installation Steps**

## 🧩 Step-by-Step (Debian/Ubuntu/RHEL)

1. **Download Universal Forwarder:**

```
wget -O splunkforwarder.tgz
"https://download.splunk.com/products/universalforwarder/releases/9.2.0/linux/spl
unkforwarder-9.2.0-xxxxxxx-Linux-x86_64.tgz"
```

2. **Extract and Install:**

```
tar -xvzf splunkforwarder.tgz
sudo mv splunkforwarder /opt/splunkforwarder
cd /opt/splunkforwarder/bin
sudo ./splunk start --accept-license
```

3. **Set Admin Credentials and Configure Forwarding:**

```
sudo ./splunk enable boot-start
sudo ./splunk add forward-server <indexer-ip>:9997
sudo ./splunk add monitor /var/log
```

**Check if it's sending logs:**

```
splunk list forward-server
```

---

## 🐳 Splunk in Docker

To run Splunk in Docker:

1. **Pull and Run:**

```
docker pull splunk/splunk:latest
docker run -d --name splunk \
  -p 8000:8000 -p 8088:8088 -p 9997:9997 \
  -e SPLUNK_START_ARGS="--accept-license" \
  -e SPLUNK_PASSWORD=YourPassword \
  splunk/splunk:latest
```

Access the UI at: http://localhost:8000

---

## ☸️ **Splunk on Kubernetes (Helm-based)**

### 🛠️ **Prerequisites:**

- Helm 3+
- Kubernetes (kind, minikube, EKS, etc.)

1. **Add Helm Repo:**

```
helm repo add splunk https://splunk.github.io/splunk-helm-chart
helm repo update
```

2. **Install Splunk Enterprise:**

```
helm install my-splunk splunk/splunk-enterprise \
  --set splunk.password='YourPassword'
```

Use kubectl get svc to get the NodePort.

---

## 🛠️ **Install Splunk as a Service (Systemd)**

**For both Splunk and Universal Forwarder:**

```
sudo /opt/splunk/bin/splunk enable boot-start
```

**To start Splunk as a systemd service:**

```
sudo systemctl start Splunkd
sudo systemctl enable Splunkd
```

---

## 🔥 Firewall Settings (UFW / firewalld)

**For Ubuntu (UFW):**

```
sudo ufw allow 8000/tcp    # Web UI
sudo ufw allow 8088/tcp    # HEC (HTTP Event Collector)
sudo ufw allow 9997/tcp    # Universal Forwarder
sudo ufw reload
```

**For RHEL/CentOS (firewalld):**

```
sudo firewall-cmd --zone=public --add-port=8000/tcp --permanent
sudo firewall-cmd --zone=public --add-port=8088/tcp --permanent
sudo firewall-cmd --zone=public --add-port=9997/tcp --permanent
sudo firewall-cmd --reload
```

---

## Splunk vs. Alternatives

Splunk stands out with its **enterprise-grade features**, ease of use, real-time monitoring, and extensive support. It is suitable for large-scale environments where advanced analytics, security, and scalability are required. Compared to alternatives like **ELK Stack** and **Graylog**, Splunk offers more out-of-the-box features and premium support, making it ideal for large organizations. However, for smaller teams or less complex systems, other tools like the ELK Stack or **Datadog** might be more cost-effective.

---

**When Splunk is Helpful in DevOps:**

- **Centralized Log Management:** Collects logs from tools like Jenkins, Docker, and Kubernetes.
- **Real-Time Monitoring:** Monitors servers and applications in real-time.
- **Alerting for Issues:** Sends alerts for failures (e.g., failed build, high CPU usage).
- **CI/CD Pipeline Monitoring:** Tracks build and deployment success or failure.
- **Troubleshooting:** Helps identify root causes by analyzing logs and metrics.
- **Security Monitoring:** Detects unauthorized access or suspicious activities.
- **Performance Optimization:** Monitors and optimizes system performance.
- **Microservices Monitoring:** Tracks the health of microservices in a distributed system.

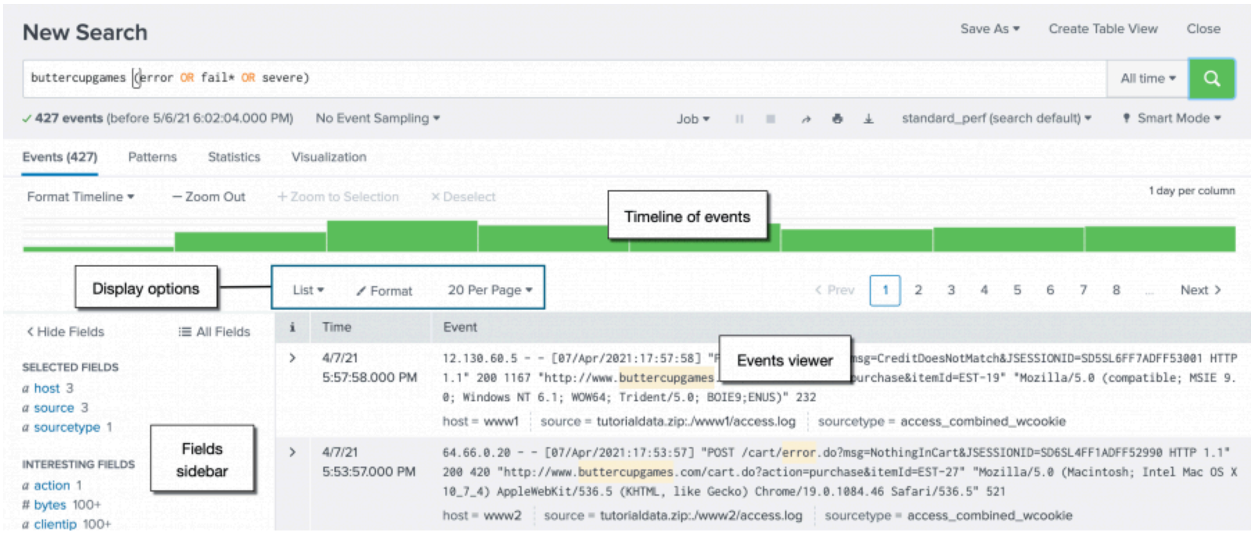
**When Splunk Might Not Be Necessary:**

- **Smaller Teams/Projects:** If the data volume is small, simpler tools like ELK Stack may suffice.
- **Budget Constraints:** Splunk can be expensive for smaller teams with limited resources.
- **Less Complex Systems:** For simple infrastructures, less advanced tools may be more appropriate.

## Conclusion

Splunk is highly beneficial for large, complex DevOps environments where real-time monitoring, advanced analytics, and security are critical. For smaller teams or projects, simpler tools like ELK Stack might be more cost-effective and easier to set up.

## Splunk UI



```
buttercupgames (error OR fail* OR severe)
```

✓ 427 events (before 5/6/21 6:02:04.000 PM)

| i | Time | Event |
|---|---|---|
| > | 4/7/21 5:57:58.000 PM | 12.130.60.5 - - [07/Apr/2021:17:57:58] "P msg=CreditDoesNotMatch&JSESSIONID=SD5SL6FF7ADFF53001 HTTP 1.1" 200 1167 "http://www.buttercupgames urchase&itemId=EST-19" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)" 232 host = www1 source = tutorialdata.zip:./www1/access.log sourcetype = access_combined_wcookie |
| > | 4/7/21 5:53:57.000 PM | 64.66.0.20 - - [07/Apr/2021:17:53:57] "POST /cart/error.do?msg=NothingInCart&JSESSIONID=SD6SL4FF1ADFF52990 HTTP 1.1" 200 420 "http://www.buttercupgames.com/cart.do?action=purchase&itemId=EST-27" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 521 host = www2 source = tutorialdata.zip:./www2/access.log sourcetype = access_combined_wcookie |

## Splunk Commands

```
# 🔧 General DevOps Logs
index=devops_logs
index=devops_logs log_level=ERROR OR log_level=CRITICAL
index=devops_logs log_level=ERROR | stats count by error_message
index=devops_logs log_level=ERROR | timechart count by host
index=devops_logs | top host
index=devops_logs earliest=-15m@m latest=now
index=devops_logs service=backend_app
```

```
index=devops_logs environment=staging
index=devops_logs | timechart span=1h count | delta count AS delta_count

# 🚀 CI/CD Tools (Jenkins, GitHub Actions, GitLab)
index=jenkins_logs "BUILD FAILURE"
index=jenkins_logs | stats avg(duration) by job_name
index=github_actions_logs status=failure
index=gitlab_logs pipeline_status=failed
index=ci_cd_logs "deployment successful"

# ☸️ Kubernetes
index=kubernetes_logs message="CrashLoopBackOff"
index=kubernetes_logs namespace=production
index=kubernetes_logs | stats count by pod_name restart_count | where
restart_count > 3
index=kubernetes_metrics metric_name="node_cpu_utilization" | timechart
avg(value) by node

# 🐳 Docker
index=docker_logs status=exited
index=docker_logs container_name=my_app_container

# 🛠️ Terraform
index=terraform_logs message="Apply complete"
index=terraform_logs log_level=ERROR

# 🛡️ Security Monitoring (DevSecOps)
index=security_logs action=failed user=*
index=security_logs status=denied

# 📊 System Metrics
index=system_metrics metric=cpu | timechart avg(usage_percent) by host
index=system_metrics metric=memory | timechart avg(used_percent) by host

# 💬 ChatOps / Slack
```

```
index=slack_logs "alert triggered"

# 🔧 General DevOps Log Exploration
index=* log_level=ERROR OR log_level=CRITICAL
index=* "Exception" OR "StackTrace" OR "failed"
index=* | stats count by sourcetype
index=* | stats count by source, host
index=* | timechart span=1h count by source
index=* | top error_message limit=10

# 📦 Application Performance Monitoring
index=app_logs response_time=*
| timechart avg(response_time) by host

index=app_logs http_status=500
| stats count by uri_path

index=app_logs
| stats avg(response_time) as avg_time by uri_path
| sort - avg_time

# 🌐 Nginx / Apache Logs
index=web_logs sourcetype=nginx:access
index=web_logs sourcetype=access_combined status>=500
index=web_logs | top uri_path
index=web_logs | stats count by status

# 🌍 Load Balancer (HAProxy / ALB / ELB)
index=lb_logs backend_status=503
index=lb_logs | timechart count by backend
index=elb_logs "Target.ResponseTime" | timechart avg(Target.ResponseTime)

# ☁️ Cloud Services (AWS / Azure / GCP)
index=aws_cloudwatch_logs message="throttling"
index=aws_billing usage_type=EC2 | timechart sum(usage_amount)
```

```
index=gcp_logs severity=ERROR
index=azure_logs operationName="Create Virtual Machine"

# ☸️ Kubernetes (Extra)
index=kube_logs "OOMKilled"
index=kube_logs container_state=terminated reason=Error
index=kube_events event_type=Warning | top reason
index=kube_metrics pod_cpu_usage | timechart avg(usage) by pod

# 🐳 Docker (Extra)
index=docker_logs event=die
index=docker_logs "OutOfMemoryError"
index=docker_logs | stats count by container_name

# 🛠️ Terraform (Extra)
index=terraform_logs "plan has been saved"
index=terraform_logs | stats count by resource_type

# 🧪 Testing / Quality Gates (Selenium, SonarQube)
index=selenium_logs test_status=FAILED
index=sonarqube_logs severity=BLOCKER
index=sonarqube_logs | stats count by rule_name

# 🔐 Security Monitoring
index=auth_logs action=failed
index=auth_logs user=root
index=* "sudo" AND ("fail" OR "denied")
index=* "unauthorized access" OR "invalid credentials"
index=* "nmap" OR "port scan detected"

# 🔎 Service Health Checks
index=uptime_logs status!=200
index=monitoring_logs service_status=DOWN
index=uptime_logs | stats count by service_name, status
```

```
# 📡 Network Monitoring
index=network_logs protocol=tcp
index=network_logs bytes_out>1000000
index=firewall_logs action=blocked
index=firewall_logs | top src_ip

# 📁 File System / Disk Space
index=system_logs "No space left on device"
index=system_metrics disk_usage | stats max(usage_percent) by mount_point

# 🧮 CI/CD Pipeline Metrics
index=ci_logs pipeline_stage="build"
index=ci_logs job_status=failed
index=ci_logs | timechart count by pipeline_stage

# 📬 Notification Systems (Email, Slack, PagerDuty)
index=notifications message="incident created"
index=email_logs status=bounced
index=slack_logs | stats count by channel_name

# 🧠 Advanced Error Investigation
index=* log_level=ERROR
| transaction session_id maxpause=10m
| table session_id, duration, error_message, host

index=* "NullPointerException" OR "OutOfMemoryError"
| stats count by host, source

# 🔁 Compare Two Deployments (Before/After)
index=deploy_logs "version=1.2.0" OR "version=1.3.0"
| rex "version=(?<app_version>[0-9\.]+)"
| stats count by app_version

# 🎯 Correlation: CI/CD + Error After Deploy
```

```
(index=ci_cd_logs "deployment successful") OR (index=app_logs
log_level=ERROR)
| transaction correlation_id maxspan=1h
| table correlation_id, _time, source, log_level

# 👣 User Journey Analysis
index=access_logs uri_path=*
| transaction user_id maxpause=30m
| table user_id, uri_path, duration

# ⏱️ Slow API Calls
index=api_logs response_time>3000
| sort - response_time
| table uri_path, response_time, host

# 🧬 Anomaly Detection
index=metrics_logs
| timechart span=5m avg(cpu_usage) as avg_cpu
| anomalydetection avg_cpu

# 📦 Release Monitoring
index=release_logs "release started" OR "release completed"
| transaction release_id
| eval status=if(searchmatch("completed"), "Success", "In Progress")
| table release_id, status, duration

# 📊 Custom Dashboard: API Errors by Country
index=api_logs log_level=ERROR
| iplocation client_ip
| stats count by Country

# 🛡️ DevSecOps: Vulnerability Scan Summary (Trivy, etc.)
index=trivy_logs severity=CRITICAL OR severity=HIGH
| stats count by vulnerability_id, severity, target
```

```
# 📅 Weekly Error Trend
index=app_logs log_level=ERROR
| timechart span=1d count as daily_errors

# 🧪 Test Flakiness Detection (CI Logs)
index=ci_logs test_status=FAILED
| stats count by test_case
| where count > 2

# ⚠️ Disk Almost Full Alerts
index=system_metrics metric=disk_usage
| where usage_percent > 85
| stats max(usage_percent) by host, mount_point

# 🔁 HTTP Redirect Loops
index=access_logs status=301 OR status=302
| transaction session_id maxevents=10
| where eventcount > 5

# 🕵️‍♀️ Privilege Escalation Detection
index=auth_logs "sudo" OR "su root"
| stats count by user, host

# 🕸️ DNS & Network Troubleshooting
index=dns_logs query_type=A OR query_type=AAAA
| top queried_domain

index=netflow_logs bytes_out>1000000
| stats sum(bytes_out) by src_ip, dest_ip

# 🎛️ Jenkins Job Run Times
index=jenkins_logs
| stats avg(duration) as avg_duration by job_name
| sort - avg_duration
```

```
# 🔄 Auto-Restarted Services (watchdog or systemd)
index=system_logs "watchdog" OR "systemd"
| stats count by service_name

# 🔓 Audit Logs (who changed what)
index=audit_logs action="modify"
| table user, object_changed, time, old_value, new_value
```

---

**Splunk Interview Q&A**

1. **What is Splunk?**
   - Splunk is a platform for searching, monitoring, and analyzing machine-generated data through a web interface.
2. **Key Components of Splunk:**
   - **Splunk Indexer:** Stores and processes data.
   - **Splunk Search Head:** Allows searching and visualization.
   - **Splunk Forwarder:** Sends data to the indexer.
   - **Splunk Deployment Server:** Manages configurations across Splunk environments.
3. **What is a Splunk Forwarder?**
   - A lightweight tool for collecting and sending logs to the Splunk Indexer.

4. **How does Splunk handle large volumes of data?**
    - It uses time-series indexing and distributes data across multiple indexers for scalability.
5. **Splunk Free vs. Splunk Enterprise:**
    - **Splunk Free** is a limited version with no clustering or advanced features.
    - **Splunk Enterprise** offers full features including clustering and distributed search.