

# pgAudit Overview

David Steele  
Crunchy Data

Crunchy Storm  
November 28, 2018



# About David

- Principal Architect at Crunchy Data, the Trusted Open Source Enterprise PostgreSQL Leader.
- Actively developing with PostgreSQL since 1999.
- PostgreSQL Contributor.
- Primary author of pgBackRest and co-author of pgAudit.

# Why pgAudit?

- The goal of the PostgreSQL Audit extension (pgAudit) is to provide PostgreSQL users with capability to produce audit logs often required to comply with government, financial, or ISO certifications.
- Organizations may also have internal requirements that can be satisfied with pgAudit.
- Can also be used for detailed debugging, metrics, and monitoring.

# Example (log\_statement = all)

- User statement:

---

```
DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;
```

---

- What gets logged:

---

```
LOG: statement: DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;
```

---

# Example (log\_statement = all)

- User statement:

---

```
DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;
```

---

- What gets logged:

---

```
LOG: statement: DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;
```

---

# Example (log\_statement = all)

- User statement:

---

```
DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;
```

---

- What gets logged:

---

```
LOG: statement: DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;
```

---

# Example (pgAudit)

- User statement:

---

```
DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;
```

---

- What gets logged:

---

```
AUDIT: SESSION,33,1,FUNCTION,DO,,, "DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;"  
AUDIT: SESSION,33,2,DDL,CREATE TABLE,TABLE,public.important_table,CREATE TABLE important_table (id INT)
```

---

# Example (pgAudit)

- User statement:

---

```
DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;
```

---

- What gets logged:

---

```
AUDIT: SESSION,33,1,FUNCTION,DO,,, "DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;"  
AUDIT: SESSION,33,2,DDL,CREATE TABLE,TABLE,public.important_table,CREATE TABLE important_table (id INT)
```

---



# Example (pgAudit)

- User statement:

---

```
DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;
```

---

- What gets logged:

---

```
AUDIT: SESSION,33,1,FUNCTION,DO,,, "DO $$  
BEGIN  
    EXECUTE 'CREATE TABLE import' || 'ant_table (id INT)';  
END $$;"  
AUDIT: SESSION,33,2,DDL,CREATE TABLE,TABLE,public.important_table,CREATE TABLE important_table (id INT)
```

---

# Stability

- Stability is one of the primary goals of pgAudit.
- Each release is maintained on a separate branch like PostgreSQL (e.g. REL\_11\_STABLE).
- Only bug fixes are back-patched.

# The Future

- Add new logging class MISC\_SET.
- Add user settable comment field.

# Questions?

website: `http://www.pgaudit.org`

email: `david@crunchydata.com`

github: `https://github.com/pgaudit/pgaudit`