

MLflow Cheatsheet

Server Connection & Setup

```
mlflow server --backend-store-uri sqlite:///
mlflow.db --default-artifact-root ./artifacts --host
0.0.0.0 --port 5000
```

Start MLflow tracking server with SQLite backend

```
mlflow server --backend-store-uri postgresql://
user:password@localhost/mlflow --default-artifact-
root s3://bucket/path
```

Start MLflow tracking server with PostgreSQL backend and S3 artifact storage

```
export MLFLOW_TRACKING_URI=http://
localhost:5000
```

Set MLflow tracking server URI in environment

```
mlflow.set_tracking_uri('http://localhost:5000')
```

Set MLflow tracking server URI in Python code

Experiment Management

```
mlflow experiments create --experiment-name
<name>
```

Create a new experiment

```
mlflow experiments list
```

List all experiments

```
mlflow experiments search --filter "name='my-
experiment'"
```

Search for experiments by name

```
mlflow.set_experiment('my-experiment')
```

Set active experiment in Python

```
mlflow experiments delete --experiment-id <id>
```

Delete an experiment by ID

Run Management

```
mlflow run <project_uri>
```

Run a project from a URI or local path

```
mlflow run . --experiment-name my-experiment
```

Run current directory project in a specific experiment

```
mlflow run . -P alpha=0.5
```

Run with a parameter

```
mlflow.start_run()
```

Start a new run in Python

```
mlflow.end_run()
```

End current run in Python

```
mlflow runs list --experiment-id <id>
```

List runs for an experiment

```
mlflow runs delete -r <run_id>
```

Delete a run by ID

Tracking

```
mlflow.log_param('learning_rate', 0.01)
```

Log a parameter in Python

```
mlflow.log_metric('accuracy', 0.95)
```

Log a metric in Python

```
mlflow.log_metrics({'accuracy': 0.95, 'f1': 0.90})
```

Log multiple metrics in Python

```
mlflow.log_artifact('plot.png')
```

Log an artifact file in Python

```
mlflow.log_artifacts('outputs')
```

Log a directory of artifacts in Python

```
mlflow.set_tag('version', 'v1.0')
```

Set a tag in Python

Model Management

```
mlflow.sklearn.log_model(model, 'model')
```

Log scikit-learn model in Python

```
mlflow.pytorch.log_model(model, 'model')
```

Log PyTorch model in Python

```
mlflow.tensorflow.log_model(model, 'model')
```

Log TensorFlow model in Python

```
mlflow.pyfunc.load_model('runs:/<run_id>/model')
```

Load a model in Python

```
mlflow models serve -m runs:/<run_id>/model -p
1234
```

Serve a model as REST API

```
mlflow models build-docker -m runs:/<run_id>/
model -n my-model
```

Build Docker image for a model

UI and Search

```
http://localhost:5000
```

Access MLflow UI in browser

```
mlflow.search_runs(experiment_ids=['0'])
```

Search runs for an experiment in Python

```
mlflow.search_runs(filter_string="metrics.accuracy
> 0.9")
```

Search runs with metric filter in Python

```
mlflow.get_run(<run_id>)
```

Get a specific run in Python