



yaml

```
- name: List files in the repository
  run: |
    ls ${ github.workspace }
```

Copy

Lists all files in the repository. The `|` allows for multiple commands (though only one is used here). `github.workspace` is the path where the repository was checked out.

yaml

```
- run: echo "🍏 This job's status is ${ job.status }."
```

Copy

Prints the current status of the job (like "success" or "failure").

This workflow is a demonstration that shows various GitHub Actions features and context variables. It doesn't do any actual testing or deployment, but shows how to access workflow information and execute basic commands.

ⓘ Claude does not have the ability to run the code it generates yet.

Retry



Claude can make mistakes. Please double-check responses.

IA



Reply to Claude...



Claude 3.5 Sonnet Choose style ▾



Privacy - Terms

```
- run: echo "🍏 This job's status is ${ job.status }."
```

Prints the current status of the job (like "success" or "failure").

This workflow is a demonstration that shows various GitHub Actions features and context variables. It doesn't do any actual testing or deployment, but shows how to access workflow information and execute basic commands.

ⓘ Claude does not have the ability to run the code it generates yet.

Retry



Claude can make mistakes. Please double-check responses.

```
- run: echo "🍏 This job's status is ${ job.status }."
```

Prints the current status of the job (like "success" or "failure").

This workflow is a demonstration that shows various GitHub Actions features and context variables. It doesn't do any actual testing or deployment, but shows how to access workflow information and execute basic commands.

ⓘ Claude does not have the ability to run the code it generates yet.

Retry



Claude can make mistakes. Please double-check responses.

```
- run: echo "🍏 This job's status is ${ job.status }."
```

Prints the current status of the job (like "success" or "failure").

This workflow is a demonstration that shows various GitHub Actions features and context variables. It doesn't do any actual testing or deployment, but shows how to access workflow information and execute basic commands.

ⓘ Claude does not have the ability to run the code it generates yet.

📄 🌟 🗨️ Retry



Claude can make mistakes. Please double-check responses.

```
- run: echo "🍏 This job's status is ${ job.status }}."
```

Prints the current status of the job (like "success" or "failure").

This workflow is a demonstration that shows various GitHub Actions features and context variables. It doesn't do any actual testing or deployment, but shows how to access workflow information and execute basic commands.

ⓘ Claude does not have the ability to run the code it generates yet.

📄 🌟 🗨️ Retry



Claude can make mistakes. Please double-check responses.

```
- run: echo "🍏 This job's status is ${ job.status }}."
```

Prints the current status of the job (like "success" or "failure").

This workflow is a demonstration that shows various GitHub Actions features and context variables. It doesn't do any actual testing or deployment, but shows how to access workflow information and execute basic commands.

ⓘ Claude does not have the ability to run the code it generates yet.

📄 🌟 🗨️ Retry



Claude can make mistakes. Please double-check responses.

```
- run: echo "🍏 This job's status is ${ job.status }}."
```

Prints the current status of the job (like "success" or "failure").

This workflow is a demonstration that shows various GitHub Actions features and context variables. It doesn't do any actual testing or deployment, but shows how to access workflow information and execute basic commands.

ⓘ Claude does not have the ability to run the code it generates yet.

📄 🌟 🗨️ Retry



Claude can make mistakes. Please double-check responses.

```
- run: echo "🍏 This job's status is ${ job.status }}."
```

Prints the current status of the job (like "success" or "failure").

This workflow is a demonstration that shows various GitHub Actions features and context variables. It doesn't do any actual testing or deployment, but shows how to access workflow information and execute basic commands.

ⓘ Claude does not have the ability to run the code it generates yet.

📄 🌟 🗨️ Retry



Claude can make mistakes. Please double-check responses.

```
- run: echo "🍏 This job's status is ${ job.status }."
```

Prints the current status of the job (like "success" or "failure").

This workflow is a demonstration that shows various GitHub Actions features and context variables. It doesn't do any actual testing or deployment, but shows how to access workflow information and execute basic commands.

ⓘ Claude does not have the ability to run the code it generates yet.

📄 🌟 🗨️ Retry



Claude can make mistakes. Please double-check responses.

```
- run: echo "🍏 This job's status is ${ job.status }."
```

Prints the current status of the job (like "success" or "failure").

This workflow is a demonstration that shows various GitHub Actions features and context variables. It doesn't do any actual testing or deployment, but shows how to access workflow information and execute basic commands.

ⓘ Claude does not have the ability to run the code it generates yet.

📄 🌟 🗨️ Retry



Claude can make mistakes. Please double-check responses.

```
- run: echo "🍏 This job's status is ${ job.status }."
```

Prints the current status of the job (like "success" or "failure").

This workflow is a demonstration that shows various GitHub Actions features and context variables. It doesn't do any actual testing or deployment, but shows how to access workflow information and execute basic commands.