Introduction to Workflow Management Solutions for Public Health Bioinformatics

Model for Distributed Public Health Bioinformatics:
Week 2 – Closer Look at WDL Tasks and Workflows

Tuesday May 04th, 2022
Kevin G. Libuit, MS | Theiagen Genomics

Training Workshop Overview

Communication and Support

- Slack workspaces:
  - **Terra-US-PHL; #wdl-writing**
  - **StaPH-B; #workflow-management, #cromwell_noobz**
- Weekly Office Hours:
  - **Mountain Region - Friday 9-10AM (PDT)**
  - **North East Region - Friday 10-11AM (PDT)**

Training Workshop Overview

Main Course Objective

Learn how to use **workflow management systems** to **develop accessible**, **interoperable**, **and reproducible** public health bioinformatics solutions

Last Week’s Content: Introduction to Workflow Management Using WDL

Major Takeaways:

- **Workflow managers** provide a **standardized framework** for creating reproducible and interoperable **bioinformatics pipelines**, especially when **containerized software are utilized**
  - Containerized software address the challenge of complex dependency libraries for accessing bioinformatics software
- **Workflow languages** are used to describe the workflow
- **Workflow engines** are used to execute and run the workflow

Last Week’s Content: Introduction to Workflow Management Using WDL

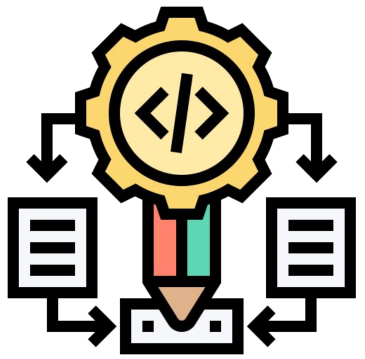
Major Takeaways:

- Workflow Description Language (WDL) - Widely-utilized workflow manager for writing reproducible and interoperable bioinformatics pipelines
  - **WDL Task** files define individual analytical modules to perform a specific bioinformatics job
  - **WDL Workflow** define the bioinformatics pipeline itself
    - Made up of individual WDL Tasks

WDL Workflows

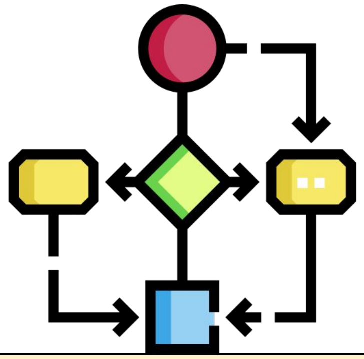
WDL Task and Workflow Files

- WDL Tasks define individual analytical modules to perform a specific bioinformatics job
- WDL Workflows define the bioinformatics pipeline itself
  - Made up of individual WDL Tasks



WDL Task

Individual step performing a single and specific bioinformatics job



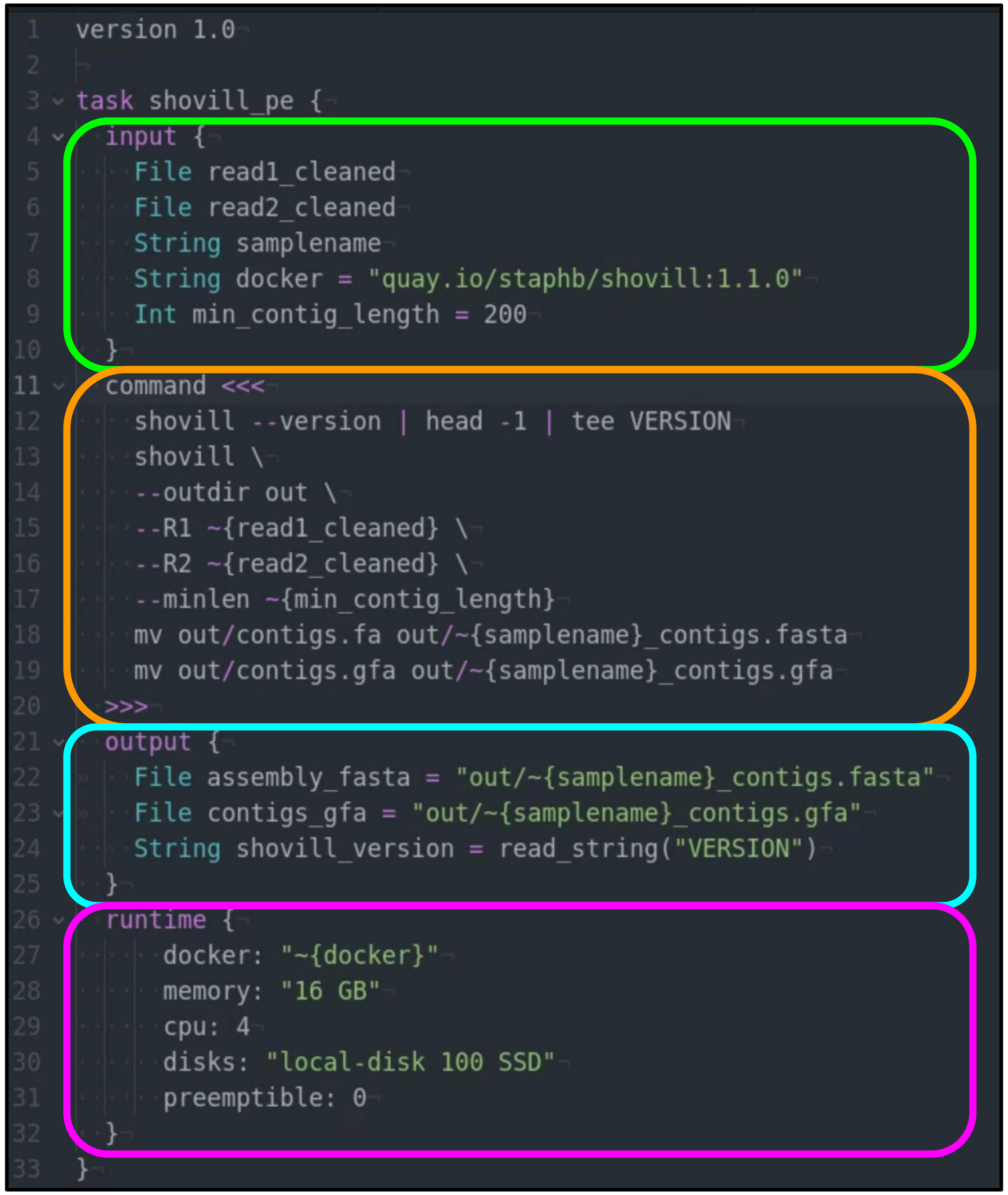
WDL Workflow

Software that strings together multiple analytical modules

WDL Task

WDL Task Elements

- Input
  - Designates the task input data and parameters
- Command
  - Defines the executables that are evaluated and executed when the task is called
- Output
  - Defines outputs of the task after a call to the task completes successfully
- Runtime
  - Defines the runtime environment in which the task in executed



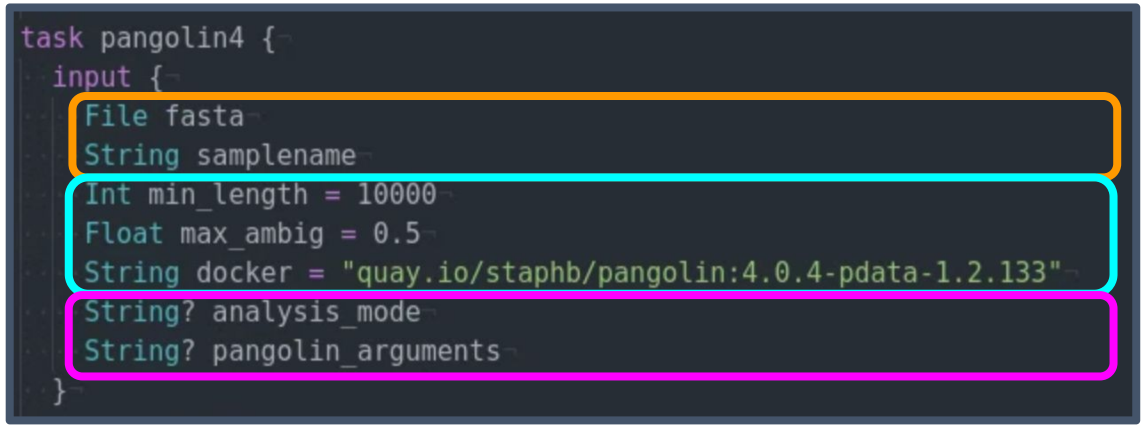
```
version 1.0

task shovill_pe {
  input {
    File read1_cleaned
    File read2_cleaned
    String samplename
    String docker = "quay.io/staphb/shovill:1.1.0"
    Int min_contig_length = 200
  }
  command <<<
    shovill --version | head -1 | tee VERSION
    shovill \
    --outdir out \
    --R1 ~{read1_cleaned} \
    --R2 ~{read2_cleaned} \
    --minlen ~{min_contig_length}
    mv out/contigs.fa out/~{samplename}_contigs.fasta
    mv out/contigs.gfa out/~{samplename}_contigs.gfa
  >>>
  output {
    File assembly_fasta = "out/~{samplename}_contigs.fasta"
    File contigs_gfa = "out/~{samplename}_contigs.gfa"
    String shovill_version = read_string("VERSION")
  }
  runtime {
      docker: "~{docker}"
      memory: "16 GB"
      cpu: 4
      disks: "local-disk 100 SSD"
      preemptible: 0
  }
}
```

task_shovill.wdl file from
Theiagen's Public Health Bacterial Genomics (PHBG) Repository

Input Section - Obligate, Optional, & Declared Inputs

Designates the **task input data** and parameters

- **Obligate Inputs (default)**
  - Required for the task to run successfully
- **Optional Inputs (`?`)**
  - Not required for the task to run successfully
- **Declared Inputs (`={declared_value}`)**
  - Default values of obligate or optional inputs; can be overridden when task is called in workflow



```
task pangolin4 {
  input {
    File fasta
    String samplename
    Int min_length = 10000
    Float max_ambig = 0.5
    String docker = "quay.io/staphb/pangolin:4.0.4-pdata-1.2.133"
    String? analysis_mode
    String? pangolin_arguments
  }
```

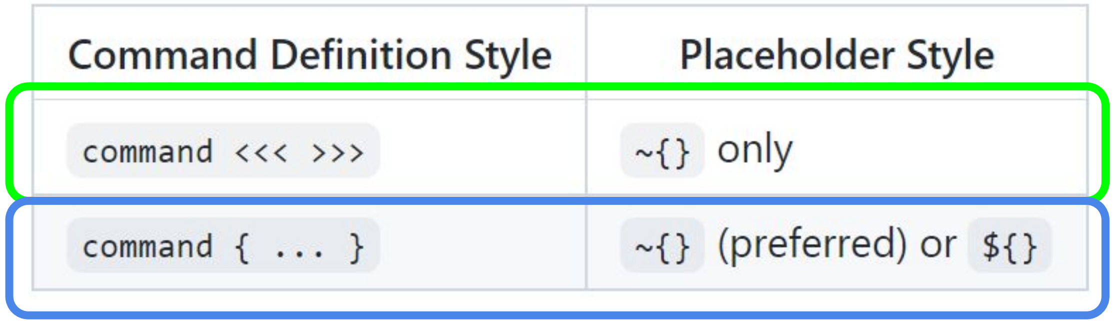
`task_taxonID.wdl` file from
Theiagen's Public Health Viral Genomics (PHVG) Repository

Command Section - Definition Style

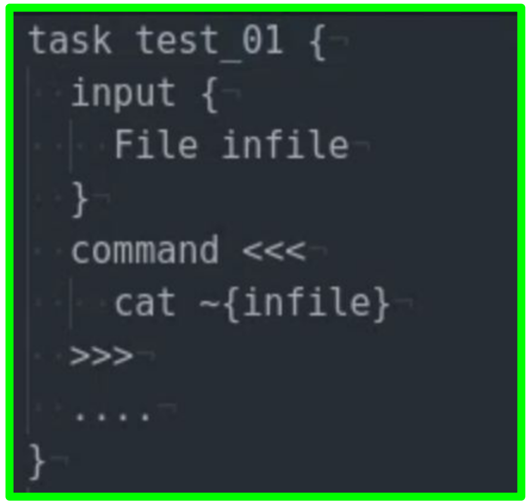
Defines the **executables that are evaluated and executed** when the task is called

- Executables within Command Section dependent on **runtime environment**
- **Two styles** available to define a command section
  - Will dictate how variable placeholders can be defined

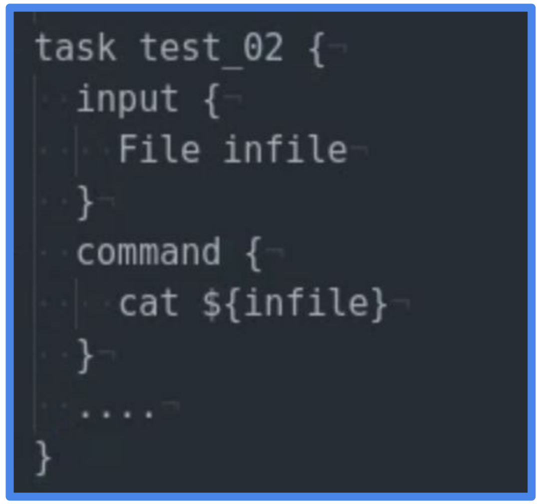
Recommend use of <<< >>> to avoid conflict with bash variables



Command Definition Style	Placeholder Style
`command <<< >>>`	`~{}` only
`command { ... }`	`~{}` (preferred) or `${}`

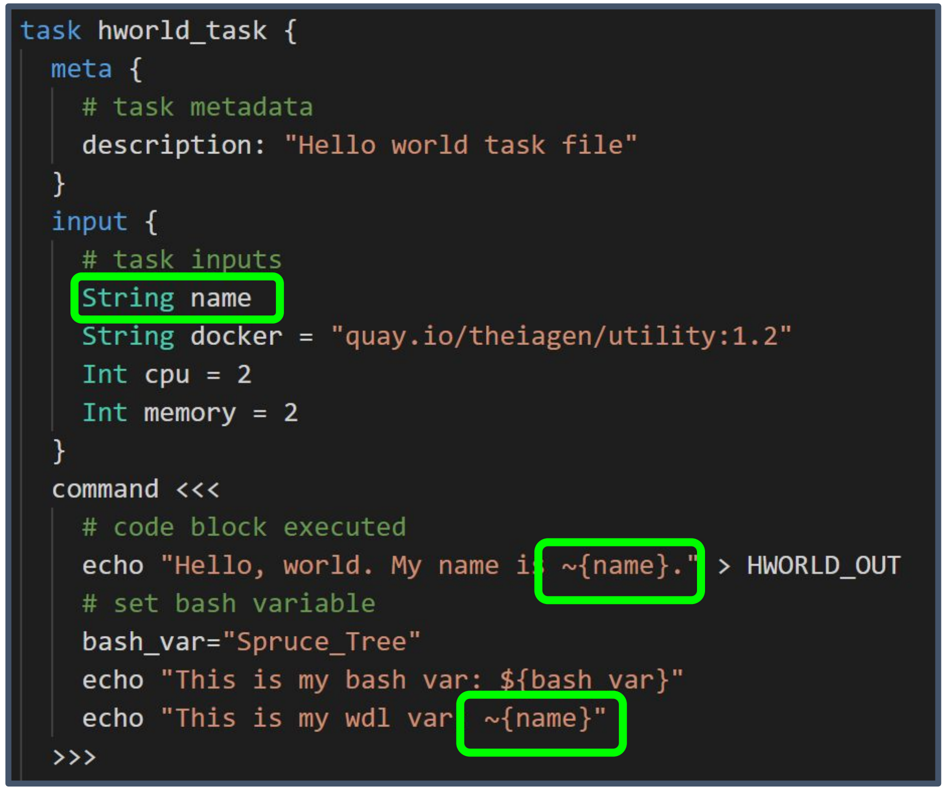


```
task test_01 {
  input {
    File infile
  }
  command <<<
    cat ~{infile}
  >>>
  ....
}
```

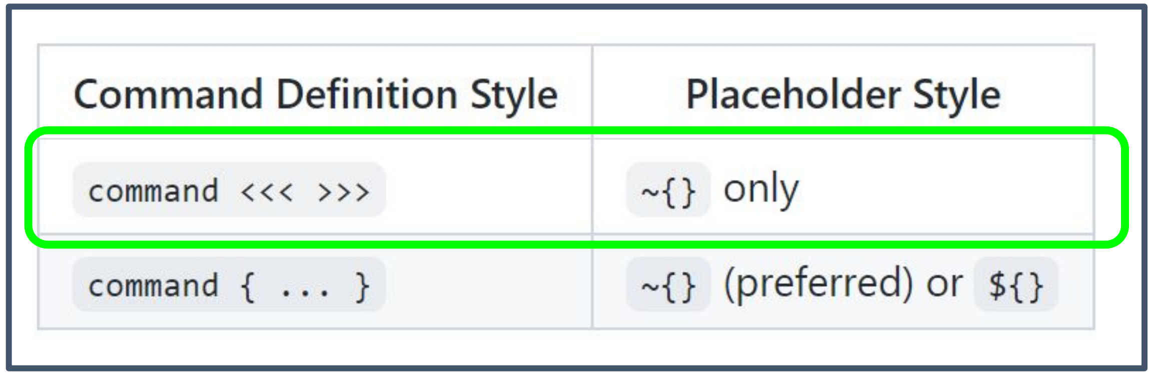


```
task test_02 {
  input {
    File infile
  }
  command {
    cat ${infile}
  }
  ....
}
```

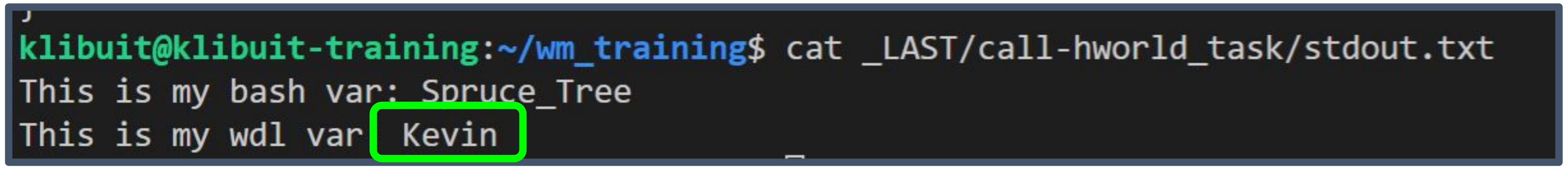
Command Section - Definition Style



```
task hworld_task {
  meta {
    # task metadata
    description: "Hello world task file"
  }
  input {
    # task inputs
    String name
    String docker = "quay.io/theiagen/utility:1.2"
    Int cpu = 2
    Int memory = 2
  }
  command <<<
    # code block executed
    echo "Hello, world. My name is ~{name}." > HWORLD_OUT
    # set bash variable
    bash_var="Spruce_Tree"
    echo "This is my bash var: ${bash_var}"
    echo "This is my wdl var: ~{name}"
  >>>
```



Command Definition Style	Placeholder Style
`command <<< >>>`	`~{}` only
`command { ... }`	`~{}` (preferred) or `${}`

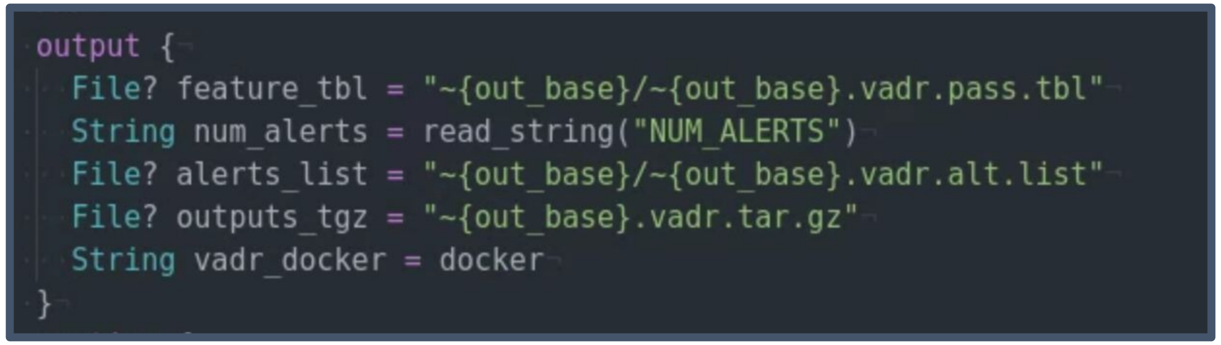


```
klibuit@klibuit-training:~/wm_training$ cat _LAST/call-hworld_task/stdout.txt
This is my bash var: Spruce_Tree
This is my wdl var: Kevin
```

Output Section - Obligate & Optional Output Values

Defines **outputs of the task** after a call to the task completes successfully

- **Obligate Outputs (default)**
  - Required for the task to complete successfully
- **Optional Outputs (?)**
  - Not required for the task to complete successfully



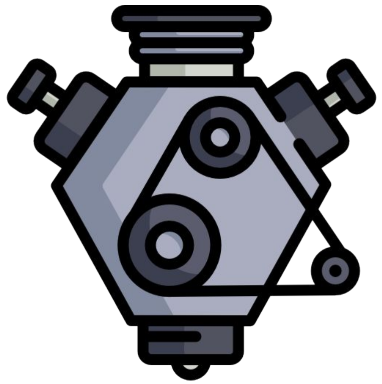
```
output {
  File? feature_tbl = "~{out_base}/~{out_base}.vadr.pass.tbl"
  String num_alerts = read_string("NUM_ALERTS")
  File? alerts_list = "~{out_base}/~{out_base}.vadr.alt.list"
  File? outputs_tgz = "~{out_base}.vadr.tar.gz"
  String vadr_docker = docker
}
```

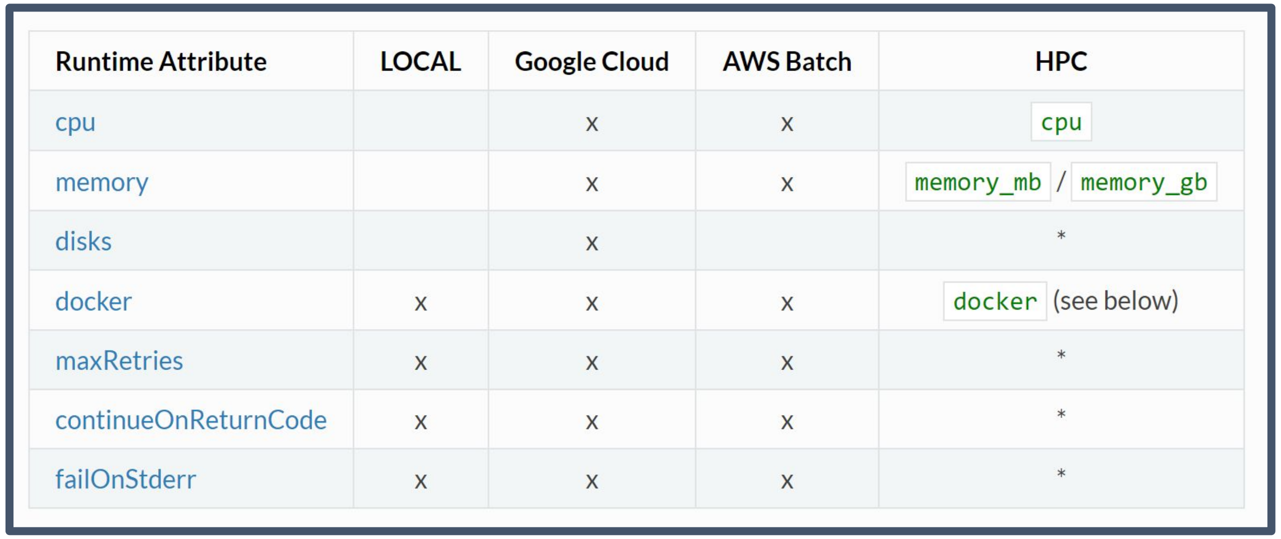
`task_ncbi.wdl` file from
Theiagen's Public Health Viral Genomics (PHVG) Repository

Runtime Section - Compute Resources

Defines the **runtime environment** in which the task in executed

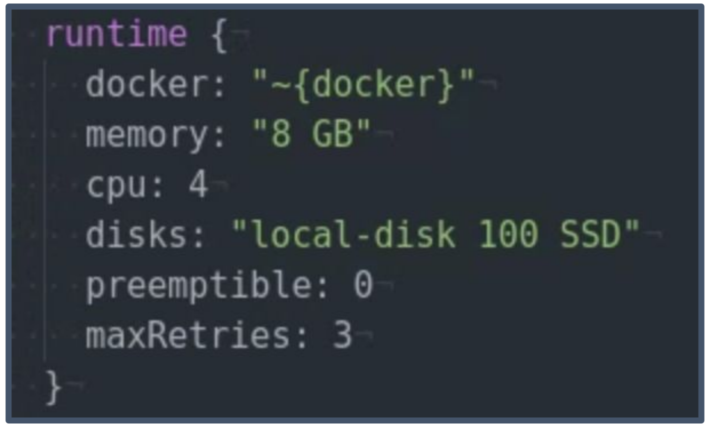
- Recognized runtime attributes depend on both **workflow engine & compute backend**





Runtime Attribute	LOCAL	Google Cloud	AWS Batch	HPC
cpu		x	x	`cpu`
memory		x	x	`memory_mb` / `memory_gb`
disks		x		*
docker	x	x	x	`docker` (see below)
maxRetries	x	x	x	*
continueOnReturnCode	x	x	x	*
failOnStderr	x	x	x	*

Cromwell Engine Runtime Attributes by Backend



```
runtime {
  docker: "~{docker}"
  memory: "8 GB"
  cpu: 4
  disks: "local-disk 100 SSD"
  preemptible: 0
  maxRetries: 3
}
```

`task_taxonID.wdl` file from
Theiagen's Public Health Viral Genomics (PHVG) Repository

Closer Look at WDL Tasks and Workflows

Major Takeaways (WDL Task Files):

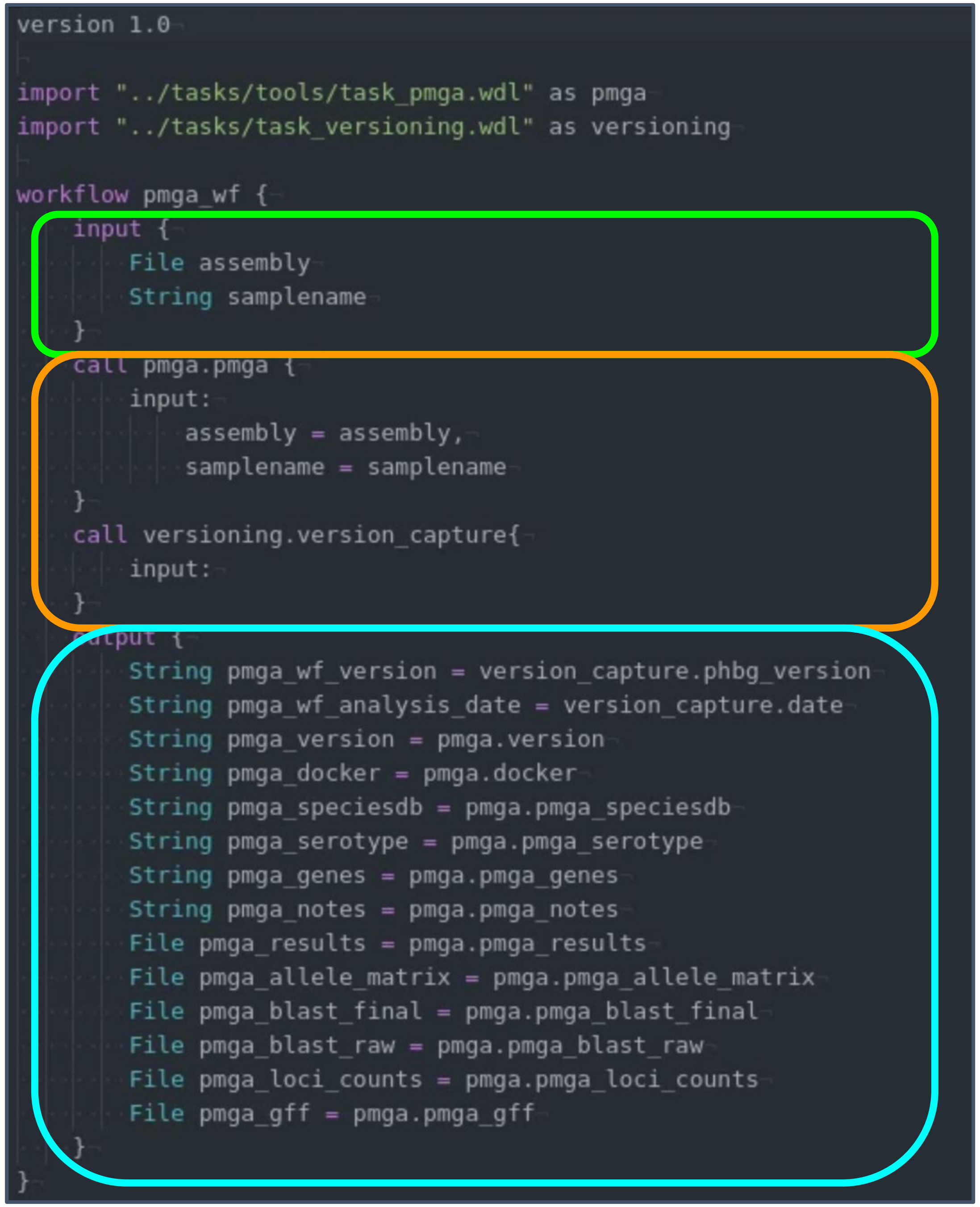
- **Input Section**
  - **Obligate Inputs:** Required for the task to run successfully
  - **Optional Inputs:** Not required for the task to run successfully
  - **Declared Inputs:** Default values of obligate or optional inputs; can be overridden when task is called in workflow
- **Command Section**
  - Two Options Available to Define a Command Element: <<< >>> or { }
- **Output Section**
  - **Obligate Outputs:** Required for the task to run successfully
  - **Optional Outputs:** Not required for the task to run successfully
- **Runtime Section**
  - Recognized runtime attributes depend on **both workflow engine & compute backend**

WDL Workflow

WDL Workflow Elements

- Input
  - Designates the workflow input data and parameters
- Call Statements
  - Defines the WDL tasks to execute as part of the WDL Workflow
- Output
  - Defines outputs of the workflow after a call to the workflow completes successfully

[`wf_pmga.wdl` file from Theiagen's Public Health Bacterial Genomics (PHBG) Repository]

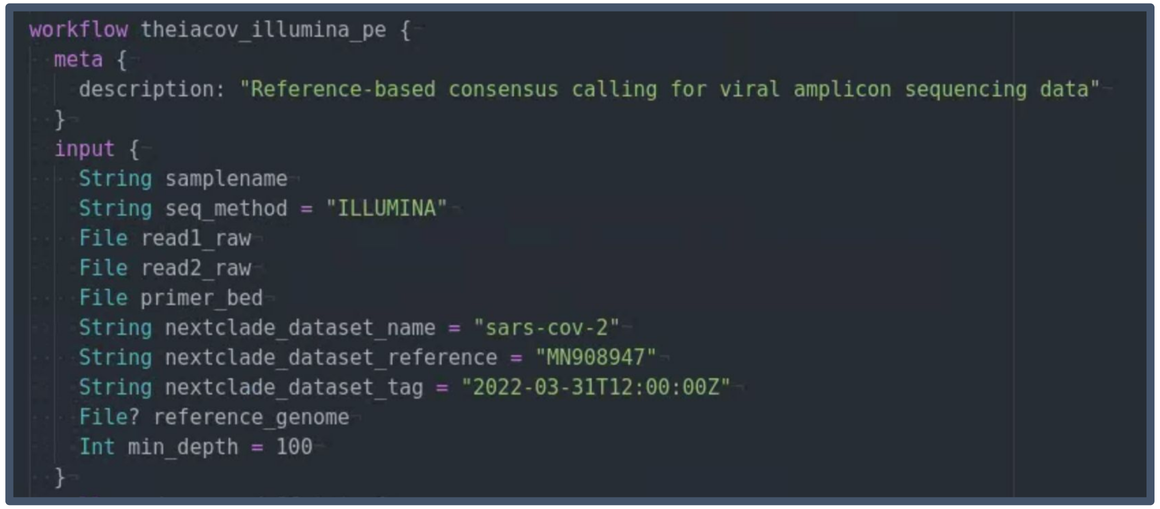


```
version 1.0

import "../tasks/tools/task_pmga.wdl" as pmga
import "../tasks/task_versioning.wdl" as versioning

workflow pmga_wf {
  input {
    File assembly
    String samplename
  }
  call pmga.pmga {
    input:
      assembly = assembly,
      samplename = samplename
  }
  call versioning.version_capture{
    input:
  }
  output {
    String pmga_wf_version = version_capture.phbg_version
    String pmga_wf_analysis_date = version_capture.date
    String pmga_version = pmga.version
    String pmga_docker = pmga.docker
    String pmga_speciesdb = pmga.pmga_speciesdb
    String pmga_serotype = pmga.pmga_serotype
    String pmga_genes = pmga.pmga_genes
    String pmga_notes = pmga.pmga_notes
    File pmga_results = pmga.pmga_results
    File pmga_allele_matrix = pmga.pmga_allele_matrix
    File pmga_blast_final = pmga.pmga_blast_final
    File pmga_blast_raw = pmga.pmga_blast_raw
    File pmga_loci_counts = pmga.pmga_loci_counts
    File pmga_gff = pmga.pmga_gff
  }
}
```

Input Section - Obligate, Optional, & Declared Inputs

Designates the **workflow input data** and parameters

- **Obligate Inputs (default)**
  - Required for the workflow to run successfully
- **Optional Inputs (`?`)**
  - Not required for the workflow to run successfully
- **Declared Inputs (`={declared_value}`)**
  - Default values of obligate or optional inputs; can be overridden when workflow is run



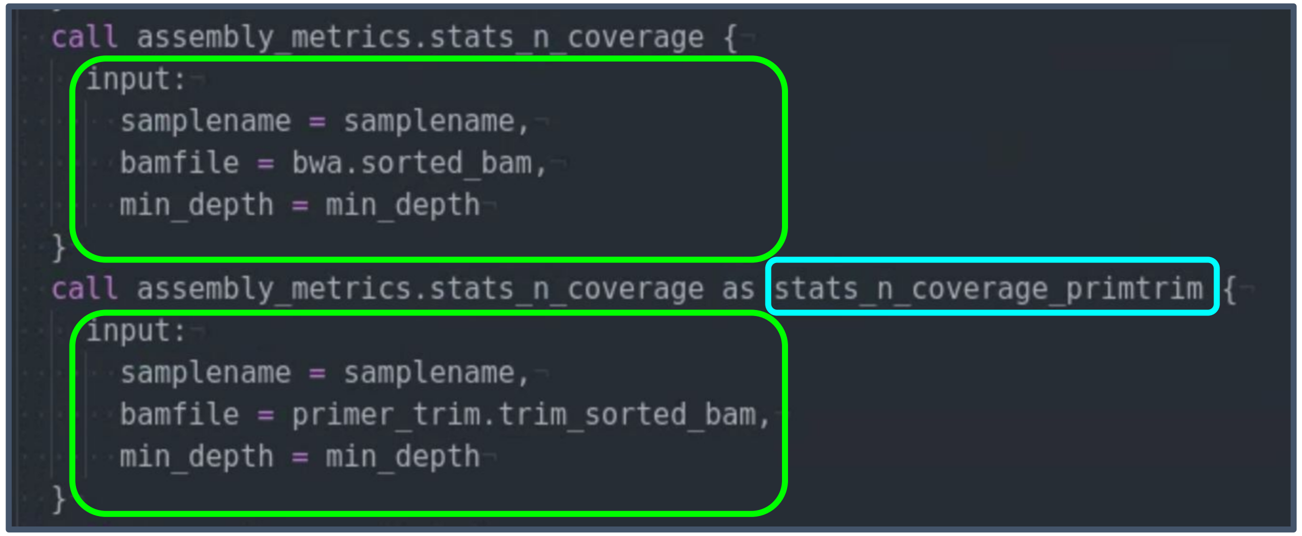
```
workflow theiacov_illumina_pe {
  meta {
    description: "Reference-based consensus calling for viral amplicon sequencing data"
  }
  input {
    String samplename
    String seq_method = "ILLUMINA"
    File read1_raw
    File read2_raw
    File primer_bed
    String nextclade_dataset_name = "sars-cov-2"
    String nextclade_dataset_reference = "MN908947"
    String nextclade_dataset_tag = "2022-03-31T12:00:00Z"
    File? reference_genome
    Int min_depth = 100
  }
```

[`wf_theiacov_illumina_pe.wdl` file from Theiagen's Public Health Viral Genomics (PHVG) Repository]()

Call Section - Task Inputs & Aliases

Defines the **WDL tasks to execute** as part of the WDL Workflow

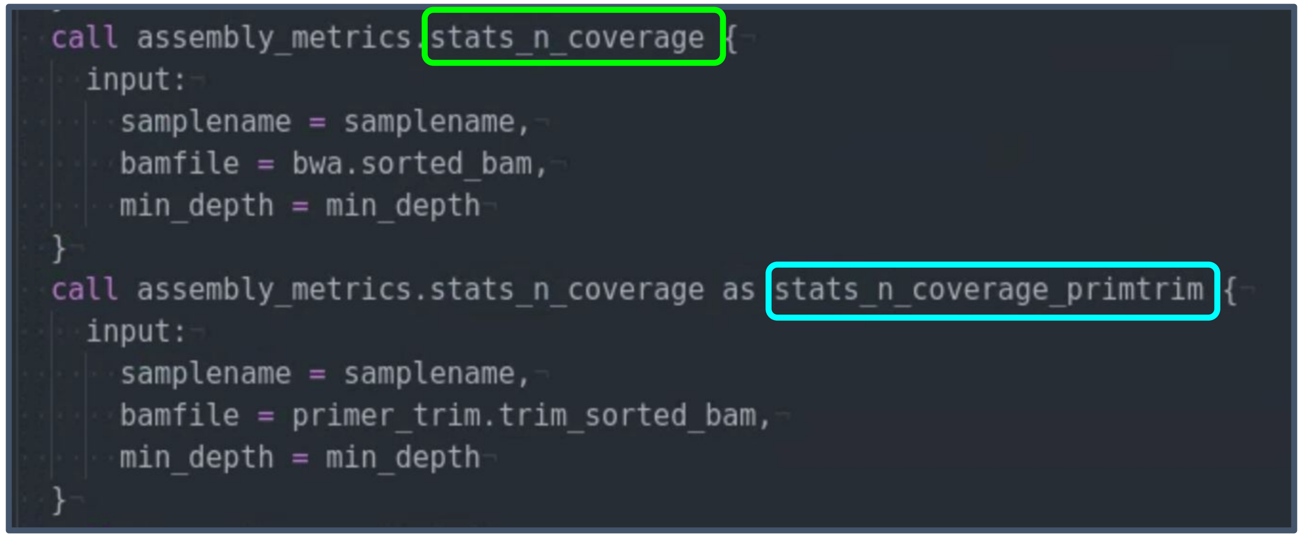
- **Task Inputs**
  - Declared for each task called – even if no input values are required
- **Task Aliases**
  - Helpful when a single task is called multiple times



```
call assembly_metrics.stats_n_coverage {
  input:
    samplename = samplename,
    bamfile = bwa.sorted_bam,
    min_depth = min_depth
}
call assembly_metrics.stats_n_coverage as stats_n_coverage_primtrim {
  input:
    samplename = samplename,
    bamfile = primer_trim.trim_sorted_bam,
    min_depth = min_depth
}
```

`wf_theiacov_illumina_pe.wdl` file from
Theiagen's Public Health Viral Genomics (PHVG) Repository

If a Task Alias is defined,
outputs are referenced using {alias}.{output} notation

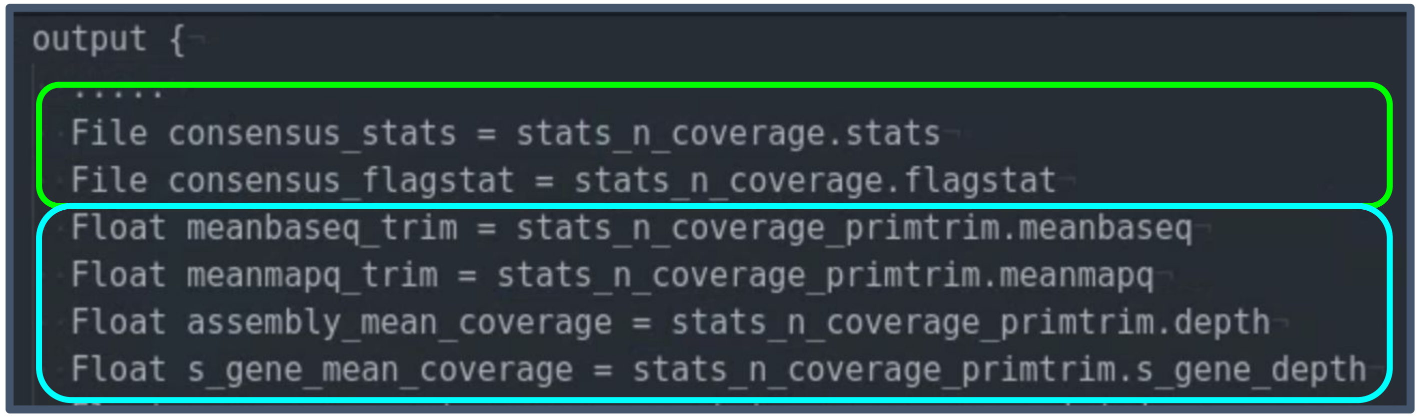


```
call assembly_metrics.stats_n_coverage {
  input:
    samplename = samplename,
    bamfile = bwa.sorted_bam,
    min_depth = min_depth
}
call assembly_metrics.stats_n_coverage as stats_n_coverage_primtrim {
  input:
    samplename = samplename,
    bamfile = primer_trim.trim_sorted_bam,
    min_depth = min_depth
}
```

`wf_theiacov_illumina_pe.wdl` file from Theiagen's Public Health Viral Genomics (PHVG) Repository

If a Task Alias is defined, **outputs are referenced using {alias}.{output} notation**

Same syntax can be used when defining a task input as a previous task's output



```
output {
  .....
  File consensus_stats = stats_n_coverage.stats
  File consensus_flagstat = stats_n_coverage.flagstat
  Float meanbaseq_trim = stats_n_coverage_primtrim.meanbaseq
  Float meanmapq_trim = stats_n_coverage_primtrim.meanmapq
  Float assembly_mean_coverage = stats_n_coverage_primtrim.depth
  Float s_gene_mean_coverage = stats_n_coverage_primtrim.s_gene_depth
```

WDL Task Files

```
task task_01 {
  input {
    File read_data
  }
  command <<<
    process_01 ~{read_data} > out_file
  >>>
  output {
    File processed_read_data = “out_file”
  }
  ...
```

```
task task_02 {
  input {
    File read_data
  }
  command <<<
    process_02 ~{read_data} > out_file
  >>>
  output {
    File processed_read_data = “out_file”
  }
  ...
```

WDL Workflow File

```
workflow workflow_01 {
  input {
    File read_data
  }
  call task_01 {
    read_data = read_data
  }
  call task_02 as alias_01{
    read_data = task_01.processed_read_data
  }
  output {
  File task_01_output = task_01.processed_read_data
  File task_02_output = alias_01.processed_read_data
  }
}
```

task_01 outputs referenced using **{task}.{output} notation**

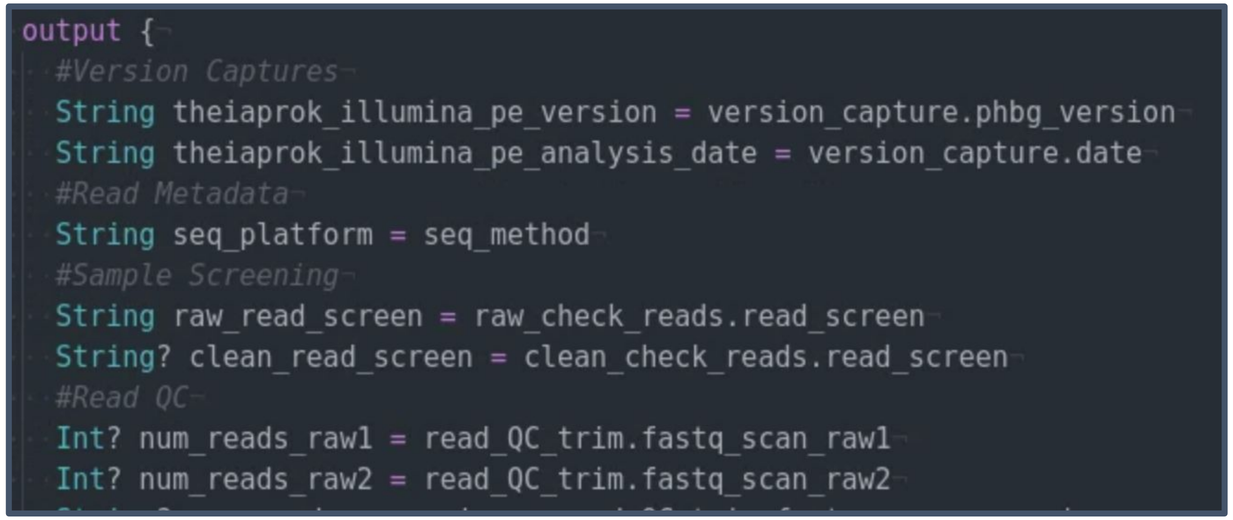
task_02 outputs referenced using **{alias}.{output} notation**

Output Section - Obligate & Optional Output Values

Defines **outputs of the workflow** after a call to the workflow completes successfully

- **Obligate Outputs (default)**
  - Required for the workflow to complete successfully

- **Optional Outputs (?)**
  - Not required for the workflow to complete successfully



```
output {
  #Version Captures
  String theiaprok_illumina_pe_version = version_capture.phbg_version
  String theiaprok_illumina_pe_analysis_date = version_capture.date
  #Read Metadata
  String seq_platform = seq_method
  #Sample Screening
  String raw_read_screen = raw_check_reads.read_screen
  String? clean_read_screen = clean_check_reads.read_screen
  #Read QC
  Int? num_reads_raw1 = read_QC_trim.fastq_scan_raw1
  Int? num_reads_raw2 = read_QC_trim.fastq_scan_raw2
```

[`wf_theiaprok_illumina_pe.wdl` file from Theiagen's Public Health Bacterial Genomics (PHBG) Repository]

Closer Look at WDL Tasks and Workflows

Major Takeaways (WDL Workflow Files):

- **Input Section**
    - **Obligate Inputs:** Required for the workflow to run successfully
    - **Optional Inputs:** Not required for the workflow to run successfully
    - **Declared Inputs:** Default values of obligate or optional inputs; can be overridden when workflow is run
- **Call Section**
    - **Task Inputs:** Declared for each task called – even if no input values are required
    - **Task Aliases:** Helpful when a single task is called multiple times
- **Output Section**
    - **Obligate Outputs:** Required for the task to run successfully
    - **Optional Outputs:** Not required for the task to run successfully

Lecture Exercise: Examining the TheiaCoV_Illumina_PE WDL Workflow

TheiaCoV_Illumina_PE:

- What are the **required inputs** for this workflow?
- What are the **optional inputs** for this workflow?
- Which tasks are given **aliases**, if any?
- Where can I find the **executed script** being run when the `consensus_call.consensus` task is called?
- What is the **default docker container image** for the `ncbi.vadr` task?

Complete this exercise during our 20m session break;
we will begin again at ______(PT)

Workflow Management Training

15m to complete Exercise 1.1

We will begin again at _____AM (PT)

Workflow Management Training

10m to begin Exercise Part 2

We will regroup again at ______AM (PT)