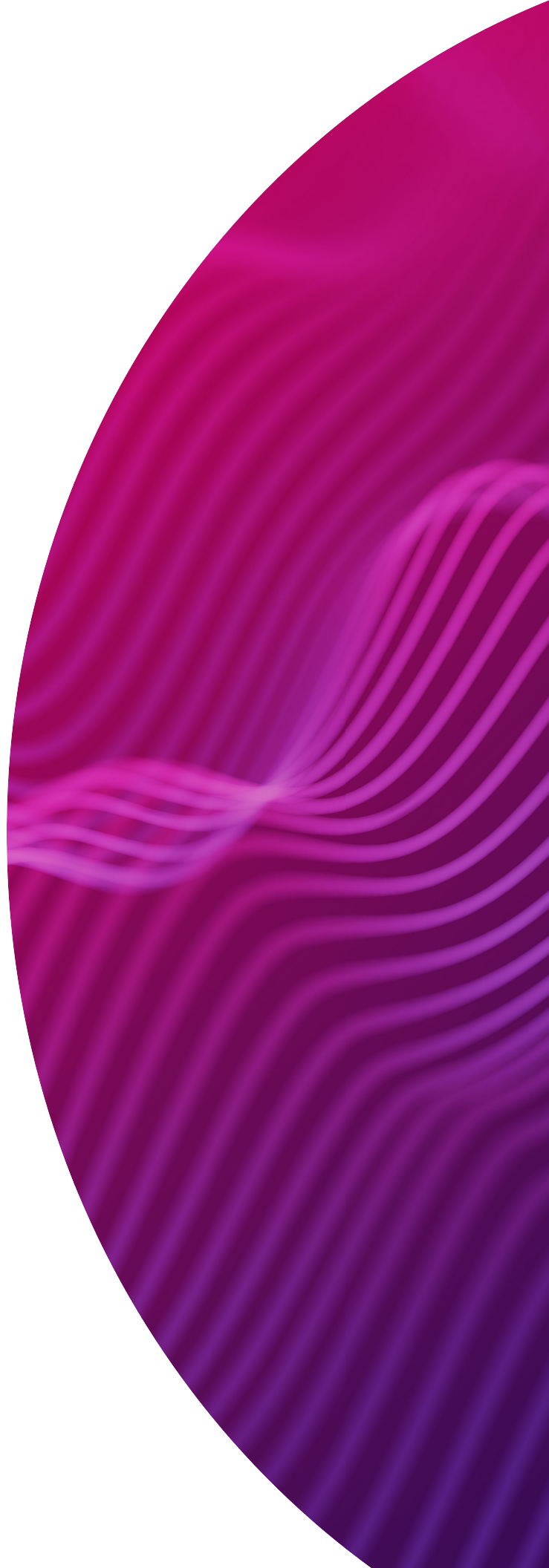




SMRT<sup>®</sup> Link  
Cloud  
reference guide  
(v11.0)



Research use only. Not for use in diagnostic procedures.

P/N 102-295-600 Version 01 (April 2022)

© 2022, PacBio. All rights reserved.

Information in this document is subject to change without notice. PacBio assumes no responsibility for any errors or omissions in this document.

You are solely responsible for properly configuring and using the SMRT Link Cloud on Amazon AWS services and otherwise taking appropriate action to secure, protect and backup your accounts and your content in a manner that will provide appropriate security and protection. You are also solely responsible for assessing whether using the SMRT Link Cloud on Amazon AWS services will meet your regulatory or other legal obligations. IN NO EVENT SHALL PACBIO BE LIABLE TO ANY USER OF OUR SERVICES OR ANY OTHER PERSON OR ENTITY FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF PROFITS, LOSS OF DATA, LOSS OF USE, OR COSTS OF OBTAINING SUBSTITUTE GOODS OR SERVICES) ARISING OUT OF THE USE, INABILITY TO USE, UNAUTHORIZED ACCESS TO OR USE OR MISUSE OF THE SMRT LINK CLOUD ON AMAZON AWS OR ANY INFORMATION CONTAINED THEREON, WHETHER BASED UPON WARRANTY, CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, EVEN IF HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR LOSSES.

Certain notices, terms, conditions and/or use restrictions may pertain to your use of PacBio products and/or third party products. Refer to the applicable PacBio terms and conditions of sale and to the applicable license terms at <http://www.pacificbiosciences.com/licenses.html>.

Trademarks:

Pacific Biosciences, the PacBio logo, PacBio, Circulomics, Omnione, SMRT, SMRTbell, Iso-Seq, Sequel, Nanobind, and SBB are trademarks of Pacific Biosciences of California Inc. (PacBio). All other trademarks are the sole property of their respective owners.

See <https://github.com/broadinstitute/cromwell/blob/develop/LICENSE.txt> for Cromwell redistribution information.

PacBio  
1305 O'Brien Drive  
Menlo Park, CA 94025  
[www.pacb.com](http://www.pacb.com)

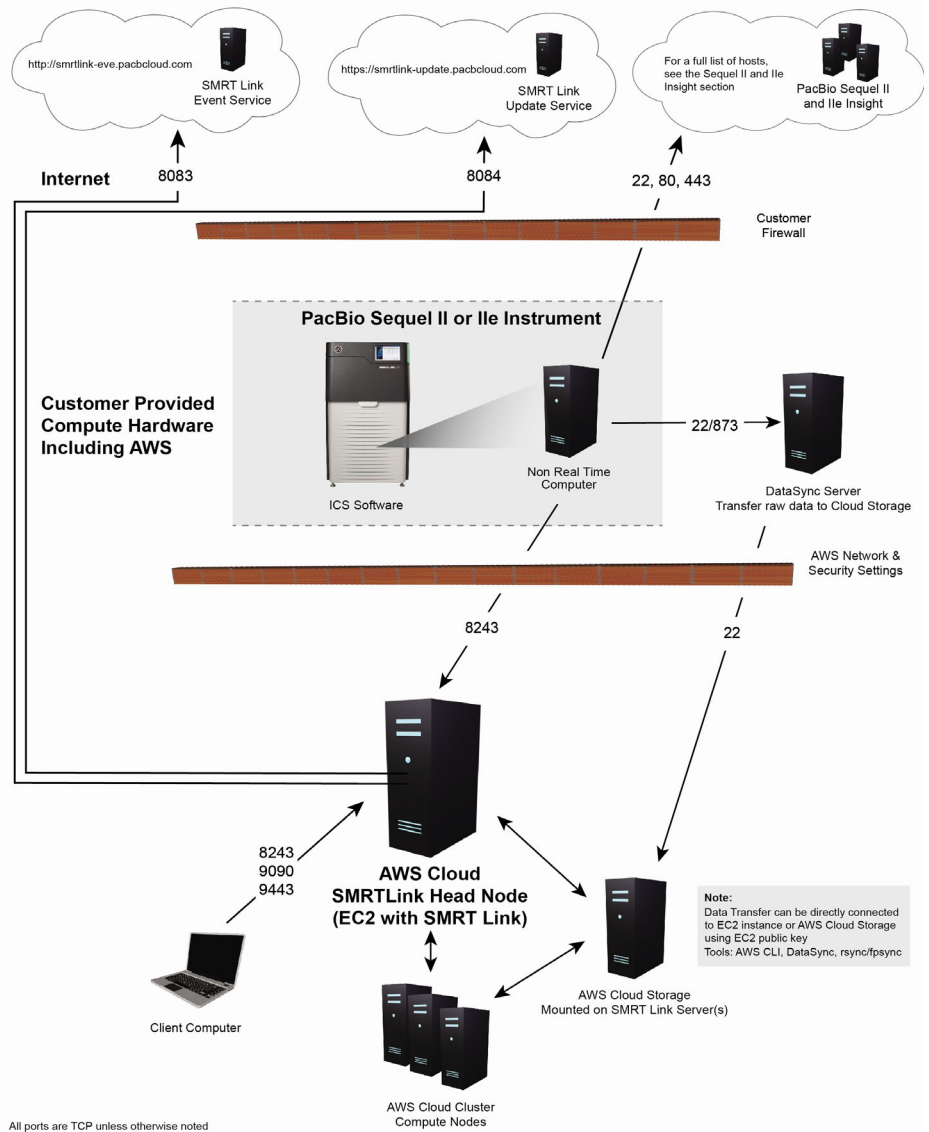
## Introduction

This document is for Customer IT or SMRT Link administrators, and describes the procedures for installing and using **SMRT Link Cloud** on Amazon AWS services. It also documents the command-line utilities provided by PacBio for use with SMRT Link Cloud, and includes a Frequently Asked Questions section.

SMRT Link Cloud works with **all** Sequel® II and Sequel Ile systems using SMRT Link v11.0.

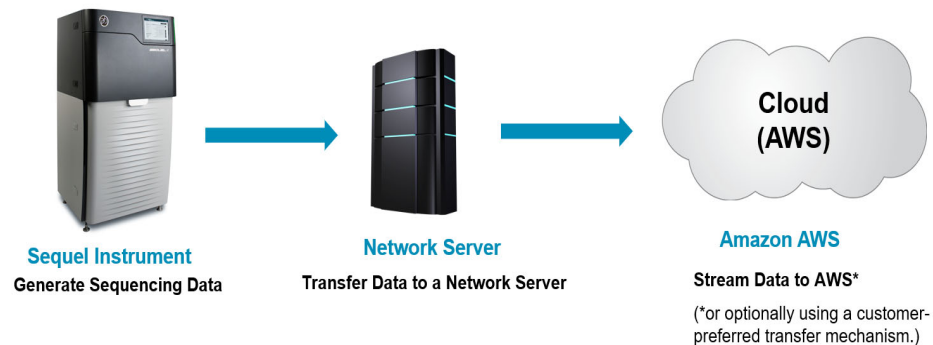
### PacBio Sequel II and Ile Network Diagram

**Recommended Configuration:** SMRT Link server is the same as the data storage server.



## Data transfer to AWS

SMRT Link Cloud provides fail-safe data streaming using a local network server to transfer the data **before** moving it to the Cloud.



## SMRT Link Cloud installation

Step	Installation summary - SMRT Link Cloud
1	Set up an Amazon Web Services account: See <a href="#">"Cloud administration: User accounts and credentials"</a> on page 5, or <a href="#">"Appendix A: Creating an AWS Amazon account"</a> on page 14.
2	Install the on-premise SMRT <sup>®</sup> Tools (or SMRT Link Cloudtools) installation, which includes <code>pbawstools</code> and <code>pbaws-efpsync</code> : See <a href="#">"SMRT Tools or SMRT Link Cloudtools software: Local/on-premise installation"</a> on page 5. <ul style="list-style-type: none"> <li>Install the <code>awscli</code> package from the Amazon, CentOS, or Ubuntu repositories: See <a href="#">"AWS CLI: Local/on-premise installation"</a> on page 5.</li> </ul>
3	Select and deploy the appropriate AWS DataSync Agent VM image (VMware/HyperV/KVM): See <a href="#">"DataSync Agent VM: Local/on-premise step"</a> on page 4. <ul style="list-style-type: none"> <li>Configure the NFS-mounted intermediary storage area and ensure the availability of SMRT Tools: See <a href="#">"Local NFS host server"</a> on page 4.</li> </ul>
4	Use <code>pbawstools</code> to provision the AWS infrastructure and deploy the SMRT Link Cloud instance: See <a href="#">"Provisioning the infrastructure and starting AWS Services"</a> on page 6.
5	Set up the local AWS DataSync server for synchronizing PacBio Data Sets to SMRT Link Cloud: See <a href="#">"Synchronizing data for use in SMRT Link Cloud"</a> on page 9. <ul style="list-style-type: none"> <li>(Optional) Try non-queued DataSync synchronization using <code>pbaws-efpsync</code>: See <a href="#">"Using pbaws-efpsync"</a> on page 10.</li> </ul>
6	Determine the URL and access the SMRT Link Cloud UI: See <a href="#">"Working with SMRT<sup>®</sup> Link Cloud"</a> on page 7. <ul style="list-style-type: none"> <li>(Optional) Accessing the CLI of the SMRT Link Cloud instance: See <a href="#">"ssh access"</a> on page 8.</li> <li>Starting, stopping, and restarting the SMRT Link Cloud web services: See <a href="#">"Stopping and restarting SMRT Link Cloud"</a> on page 8.</li> </ul>

## Prerequisites for setting up SMRT Link Cloud

### Required local network server hardware requirements

AWS DataSync VM Specifications (Minimum requirements)	
Virtual processors – Four (4) virtual processors assigned to the VM	
Disk space – 80 GB of disk space for installation of VM image and system data	
RAM – Depending on your configuration, one of the following:	
• 32 GB of RAM assigned to the VM, for tasks to transfer EC2 instance types with <u>up to 20 million files</u> .	
• 64 GB of RAM assigned to the VM, for tasks to transfer <u>more than 20 million files</u>	
Storage Requirements	
Sequel Ite System	1.5 – 2 TB (no kinetics info), 9 – 12 TB (with kinetics info)
Sequel II System	12 TB
Network (Instrument – Storage Server)	
10 GBE highly recommended, 1 GBE required	
Internet Connection	
Business-class dedicated high-speed internet connection required*	

\*Time for data transfer to AWS is highly dependent on network speed and load.

**Example:** Observed transfer speed of 100 GB/hour on a 300 Mbps connection with very light additional load.

### Software requirements - Local network server

#### 1. AWS Command Line Interface (AWS CLI)

- Configure AWS, specifying a region with credentials.
- See [“AWS CLI: Local/on-premise installation” on page 5](#) for details.

#### 2. SMRT Tools v11.0 installation

- Includes command-line tools to create, start, stop and delete SMRT Link Cloud.
- Includes command-line tools for data synchronization.
- See **SMRT Link software installation guide (v11.0)** for details.

#### 3. DataSync Agent Virtual Machine

- Enables automated data synchronization.
- See [“DataSync Agent VM: Local/on-premise step” on page 4](#) for details.

#### 4. Linux command-line tools

- rsync
- fpsync

### SMRT Link software: Cloud/internet step

Upload the SMRT Link tarball to an accessible internet location. The SMRT Link tarball needs to be accessible for cloud installation on AWS.

---

**Note:** This location must be specified as the value for `SmrtLinkSoftwareLink` when the `pbawstools create` command is executed. See [“Provisioning the infrastructure and starting AWS Services” on page 6](#) for details.

**Example location 1:**

```
'SmrtLinkSoftwareLink': 'https://pb-sl-cr-test.s3-us-west'
```

**Example location 2:**

```
'SmrtLinkSoftwareLink': 'https://amazonaws.com/current_develop_smrtlink-cleanbuild_tarball.run'
```

**Note:** SMRT Link can be accessed from **any** accessible internet location, not just from S3.

**DataSync Agent VM: Local/on-premise step**

Work with your IT department to download and deploy a VM at your site. Requirements for the on-premise VM are listed [here](#).

Depending on the hypervisor used, choose one of the corresponding agent-images:

```
https://d8vjazrbkazun.cloudfront.net/AWS-DataSync-Agent-VMWare.zip  
https://d8vjazrbkazun.cloudfront.net/AWS-DataSync-Agent-KVM.zip  
https://d8vjazrbkazun.cloudfront.net/AWS-DataSync-Agent-HyperV.zip
```

Obtain the VM IP address from your IT department. This IP address is the `--agentIP` or `-aip` to use in [“Synchronizing data for use in SMRT Link Cloud” on page 9](#).

**Local NFS host server**

The SMRT Link Cloud data transfer model uses an intermediate local NFS host server where the data from the sequencing instrument is transferred to first **before** it is synchronized with the cloud. This mechanism is intended to prevent any data loss in case of network connectivity interruption.

As part of the SMRT Link distribution, PacBio provides utilities to assist with data transfer to AWS. These should be installed on the intermediate file transfer host using the `--smrttools-only` option. The server operating system requirements are the same as for SMRT Link, but the compute hardware requirements are minimal since this host will **not** be running the SMRT Link server.

**DataSync using pbaws-efpsync: Local/on-premise step**

To use `pbaws-efpsync` for synchronizing data, `/usr/bin/rsync` and `/usr/bin/fpsync` should already be installed or be available on the on-

---

**Cloud administration: User accounts and credentials**

premise host. See [“Synchronizing data for use in SMRT Link Cloud” on page 9](#) for details.

An AWS Amazon Account is needed for installation of SMRT Link in the cloud. See [“Appendix A: Creating an AWS Amazon account” on page 14](#) for details.

**Note:** We **strongly** recommend subscribing to an AWS support plan – see [here](#) for details. Note that for production-level workload, Amazon suggests **Business**-level support.

The cloud administrator needs to enable the SMRT Link Cloud users within the organization. The Cloud administrator creates an AWS cloud-user account, AWS Access Key, and Secret Access Key that are then emailed to users in a .csv file.

The cloud administrator creates a new EC2 keypair key-name and key-name.pem file and emails those to the users. Users then store the key-name.pem file in their local ~/ .ssh directory. In addition, permissions **must** be set so that the file is read-only for the user, using `chmod 400 key-name.pem`.

### **SMRT Tools or SMRT Link Cloudtools software: Local/on-premise installation**

Install **either** SMRT Tools **or** SMRT Link Cloudtools on the on-premise host and make sure that it is added to the path. **Both** installations provide access to:

- pbawstools (See [“pbawtools” on page 15](#) for details.)
- pbaws-datasync (See [“pbaws-datasync” on page 20](#) for details.)
- pbaws-efpsync (See [“pbaws-efpsync” on page 22](#) for details.)

**Note:** SMRT Link Cloudtools includes **only** the tools needed for use with SMRT Link Cloud; SMRT Tools installs the **full** set of command-line SMRT Link tools.

#### **Installing SMRT Link Cloudtools on the on-premise host:**

```
./smrtlink_cloudtools_<version number>.run --rootdir smrtroot
```

#### **Installing SMRT Tools on the on-premise host:**

```
./smrtlink_<version number>.run --rootdir smrtlink --smrttools-only
```

### **AWS CLI: Local/on-premise installation**

Install `awscli` and make sure it is added to the path (<https://aws.amazon.com/cli/>). Alternatively, use the AWS console to configure your AWS.

---

## Configuring the AWS CLI

```
$ aws configure

AWS Access Key ID [None]: xxxxxxxxxxxx
AWS Secret Access Key [None]: xxxxxxxxxxxx
Default region name [None]: us-west-2
Default output format [None]: json
```

For more information, see [here](#).

## Provisioning the infrastructure and starting AWS Services

Use `pbawstools` to handle AWS SMRT Link resources and services. The tool lets you create, stop, start, update, and delete SMRT Link resources from your AWS instance. The tool is part of your SMRT Tools-only installation and can be found in `$SMRT_ROOT/smrtcmands/bin`.

```
$ pbawstools --help
```

## Creating a stack using `pbawstools create`

A stack is a collection of AWS resources that you can manage as a single unit. You create, update, or delete a collection of resources by creating, updating, or deleting stacks. Use the `pbawstools create` command to create a stack for the SMRT Link installation.

`pbawstools create` requires a stack name (a unique and arbitrary name), AWS account credentials, and AWS configuration parameters as input.

```
pbawstools create \
-sn <stack-name> \
-spk /path/to/.ssh/key.pem \
-akp <Key-pair name> \
-akId <Access Key ID> \
-ask <Secret Key ID> \
-acfp <or -acfpf "SmrtLinkSoftwareLink">
```

**Note:** The stack name **must** satisfy the following regular expression pattern: `[a-zA-Z][-a-zA-Z0-9]*`

To see more information about `pbawstools create`:

```
$ pbawstools create --help
```

To list the list of stacks currently created and available:

```
$ aws cloudformation list-stacks --stack-status-filter
CREATE_COMPLETE
```

To delete the stack:

```
$ pbawstools delete -sn <stack-name>
```

**Note: Everything** will be removed after deleting the stack - you will no longer have access to the data, SMRT Link and results.



---

## Working with SMRT Link Cloud

### Storing and archiving data on the cloud

AWS provides capabilities for storing and archiving data in the cloud. Refer to Amazon's AWS support information on the topic. In the current implementation, anything in Amazon Elastic File System (EFS) that is not used for more than 7 days is **automatically** archived to lower-cost archive storage and brought back to active storage automatically when accessed the next time. See [here](#) for details.

To archive the data **after** deleting an AWS SMRT Link instance, see [here](#).

### Accessing SMRT Link Cloud

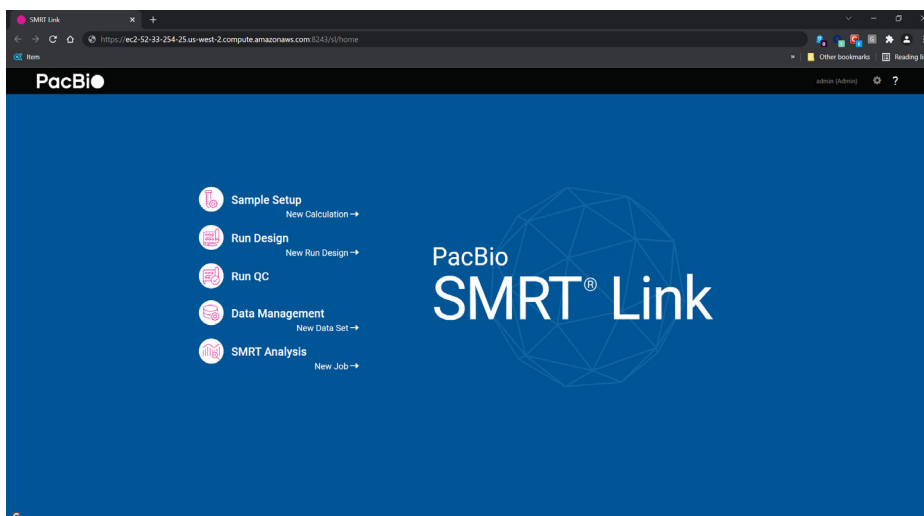
Once the stack has been created successfully, retrieve the URL to access SMRT Link Cloud:

```
$ pbawstools --quiet url -sn <stack-name>
```

#### Sample output:

```
[INFO] 2020-07-08 19:50:38,783Z [pbawstools.sl_aws run_args 699] https://ec2-52-33-254-25.us-west-2.compute.amazonaws.com:8243/sl/home
```

Copy the URL and paste it in a new Google Chrome window as shown below:



To securely log in into SMRT Link Cloud, enter your user name (`admin`) and password (your AWS Access Key ID). We recommend changing the user name and password for privacy on the URL to access SMRT Link/carbon, as in the example below:

```
https://ec2-52-33-254-25.us-west-2.compute.amazonaws.com:9443/carbon
```

After logging in, see the document **SMRT Link user guide (v11.0)** for instructions on how to use SMRT Link.

---

## ssh access, stopping and starting SMRT Link Cloud

### ssh access

**Warning:** We have **not** tested this option and **discourage** the use of `ssh`.

You can access the command line using `ssh`:

```
ssh -i /path/to/.ssh/key.pem ec2-user@ec2-100-20-135-42.us-west-2.compute.amazonaws.com
```

```
__|  ( __|_ )  
_|  /  
__|\__|__|  
Amazon Linux 2 AMI
```

```
https://aws.amazon.com/amazon-linux-2/  
4 package(s) needed for security, out of 8 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-36-135 ~]$
```

### Stopping and restarting SMRT Link Cloud

For the default SMRT Link head node, the cost of running SMRT Link Cloud in the background is about 25 cents per hour. (`r5.xlarge` is the instance that we use for the head node, in the `us-west-2` region; check the cost for your region.)

When **not** actively using SMRT Link Cloud services, you can stop them. This ensures that there are **no** head node idle costs. Once services are stopped, the SMRT Link web interface becomes inaccessible, and **no** users can create analyses or view results. (If you need constant access for multiple users, do **not** stop SMRT Link Cloud Services.) Note that your data is **not** lost if services are stopped.

Stopping services when **not** using SMRT Link Cloud also helps to accumulate Amazon Elastic File System (EFS) burst credits, which can be used for better EFS performance at lower cost.

To stop SMRT Link Cloud, run the `pbawstools stop` command on the on-premise host:

```
$ pbawstools stop -sn <stack-name> -spk <ssh-pem-key>
```

**To restart SMRT Link Cloud services using** `pbawstools start`:

Use `pbawstools start` on the on-premise host to restart the SMRT Link Cloud services:

```
$ pbawstools start -sn <stack-name> -spk <ssh-pem-key>
```

### Monitoring SMRT Link Cloud usage

Use the following procedure to find more information on who is running analyses, and their associated costs.

1. Logon to the AWS console, Batch Service.

- 
2. Click on Dashboard on the left panel and look for your `<stack-name>-DefaultQueue` in the Job queue overview table.
  3. Click on the corresponding numbers under Job States, which lists the details of the analysis jobs in those states.

It is not easy to calculate the cost for analysis jobs individually, but the **total** cost for your stack is easily calculated. You can activate billing based on user-defined tags for your AWS account [here](#). It may take up to 24 hours for this AWS billing service to get activated. Then, go to **AWS Cost Management > Cost Explorer**.

Group by the tag `STACKNAME` (all caps), which lists all the cost associated with anything run for your stack `<stack-name>`.

**Note:** IAM (Identity and Access Management) users need special access from their AWS administrators to see the billing costs.

## Synchronizing data for use in SMRT Link Cloud

The SMRT Link Cloud data transfer model uses an intermediate local NFS host server where the data from the sequencing instrument is transferred to first **before** it is synchronized with the cloud. This mechanism is intended to prevent any data loss in case of cloud connectivity interruption.

Create a directory on your mounted NFS server that contains the files to synchronize to SMRT Link Cloud. By default, the names are the following:

```
AWS_SYNC
├── TO_AWS
└── FROM_AWS
```

Data that needs to be transferred **to** SMRT Link Cloud should be in the `AWS_SYNC/TO_AWS` directory.

`AWS_SYNC/FROM_AWS` contains the data to copy back **from** SMRT Link Cloud.

The synchronization is done **automatically**. Data transfer and synchronization depends on data size and network speed. The directory structure used for synchronization with SMRT Link Cloud is as follows:

```
pacbio-root/pacbio-data
├── sync-in
└── sync-out
```

The files in `pacbio-root/pacbio-data/sync-in` are synchronized with the content in `AWS_SYNC/TO_AWS`, and the files in `pacbio-root/pacbio-data/sync-out` are synchronized with `AWS_SYNC/FROM_AWS`.

**Any** changes to the `AWS_SYNC` directory affect the content in `pacbio-root/pacbio-data`.

---

Ensure that **all** the files and folders to be synchronized have the following permissions:

- **Directories:** `$ chmod -R 755 <foldername>`
- **Files:** `$ chmod 444 <filename>`

### File formats included in the DataSync

**Everything** in the source directory will be transferred to AWS **except** files with the following extensions:

- `*.trc.h5*`
- `tmp-file-*.txt`
- `*.baz`

Run `pbaws-datasync` to synchronize your local storage with SMRT link Cloud:

```
$ pbaws-datasync create \  
-sn <stack-name> \  
-sd /path/to/AWS_SYNC/TO_AWS/ \  
-dd /path/to/AWS_SYNC/FROM_AWS/ \  
-add /pacbio-root/pacbio-data/sync-in \  
-asd /pacbio-root/pacbio-data/sync-out \  
-aip <IP-address> \  

```

**Note:** To obtain the `-aip <IP-address>`, see [“Prerequisites for setting up SMRT Link Cloud” on page 3](#).

For more information about `pbaws-datasync`:

Enter `$ pbaws-datasync create --help` (or see [“pbaws-datasync” on page 20](#).)

In case of errors when you need to recreate the DataSync step, you need to first **delete** the existing DataSync and then recreate it using `pbaws-datasync create`:

```
$ pbaws-datasync delete -sn <stack-name>
```

The DataSync jobs are queued; priority **cannot** be specified.

### Using `pbaws-efpsync`

To sync anything outside `aws datasync`, use `$ pbaws-efpsync`.

`pbaws-efpsync` has several commands to synchronize data, including `to_aws`, `from_aws` and `local`.

The `pbaws-efpsync to_aws` command allows immediate data upload to the AWS cloud. You need to define either the stack name or the AWS EC2 host to identify your SMRT Link Cloud instance. In addition, the command requires the absolute path of the `ssh pem` key file, a local source, and an AWS destination folder.

---

**Example:** Upload to an AWS cloud instance without importing a Data Set into SMRT Link, using the stack name for identification:

```
$ pbaws-efpsync to_aws \  
-sn <stack-name> \  
-spk <ssh-pem-key> \  
-s /path/to/source_dir \  
-d /path/to/aws_destination_dir
```

**Example:** Upload to an AWS cloud instance using the AWS EC2 host for identification:

```
$ pbaws-efpsync to_aws \  
-shd ec2-52-35-146-182.us-west-2.compute.amazonaws.com \  
-spk <ssh-pem-key> \  
-s /path/to/source_dir \  
-d /path/to/aws_destination_dir
```

**Example:** Upload to an AWS cloud instance with import of a Data Set into SMRT Link:

```
$ pbaws-efpsync to_aws \  
-sn <stack-name> \  
-spk <ssh-pem-key> \  
-s /path/to/local_source_dir \  
-d /path/to/aws_destination_dir \  
--import-datasets \  
--smrtlink-user admin \  
--smrtlink-password <pwd> \  
--block-for-import
```

Using `pbaws-efpsync from_aws`, you can download data from the AWS cloud instance to a local storage server. The syntax is similar to `pbaws-efpsync to_aws`. The `aws` directory is the **source**, and a local folder the **destination**.

**Example:**

```
$ pbaws-efpsync from_aws \  
-sn <stack-name> \  
-spk <ssh-pem-key> \  
-s /path/to/aws_source_dir \  
-d /path/to/local_destination_dir
```

**Note:** With `pbaws-efpsync to_aws` and `from_aws`, the source and destination directories can be **different** from those set up in [“Synchronizing data for use in SMRT Link Cloud” on page 9](#).

You can use `pbaws-efpsync local` to copy data from a local storage source to the `AWS_SYNC/TO_AWS` DataSync directory. This ensures that permissions are set correctly for synchronization.

---

## Example:

```
$ pbaws-efpsync local \  
-s /path/to/local_storage_dir \  
-d /path/to/AWS_SYNC/TO_AWS
```

## Monitoring DataSync progress

1. Logon to the AWS console and click on **Datasync services**. In **Datasync**, click on the **Tasks** link on the left side panel.
2. Click on your corresponding Task ID link. The Task, which is actually the task name, begins with your AWS stack name.
3. Click on the **History** tab, which lists the execution history.
4. Click on the execution ID link where the start time is after you rsynced the files to the `TO_AWS` source directory; this lists the status of that execution.
5. Click on **Task Logging** in the above window; it will have a link for the cloud watch log stream. Click on that link and search for that particular execution ID in the list of log files; this lists all the files transferred.

## Mechanisms used for the DataSync

Data transfer uses the AWS DataSync tool - see [here](#) for details. DataSync includes encryption and integrity validation to help the data arrive securely, intact, and ready to use. DataSync automates both the management of data transfer processes and the infrastructure required for high-performance, secure data transfer. DataSync can use **either** public service endpoints (the default) in their respective AWS Regions, or transfer data via Direct Connect or VPN:

- Transfer data via Direct Connect or VPN using private IP addresses accessible **only** from within your Virtual Private Cloud (VPC). This allows you to eliminate **all** internet access from your on-premise DataSync server, but still use DataSync for data transfers to and from AWS using Private IP addresses. See [here](#) for details.
- AWS Datasync FAQ: See [here](#) for details.

Other methods of data transfer include `rsync`, and `fpsync` via `ssh`. For additional information, see [here](#).

To copy from or to EFS: See [here](#) for details.

One of the following two mutually-exclusive DataSync methods can be used with the AWS DataSync VM.

- `pbaws-datasync create -sd <source_dir>`: Synchronizing periodically at the start of every hour, any files changed in the source directory during the last hour are synchronized to AWS. This method creates two DataSync tasks, one `to_aws` and one `from_aws` between a pair of source and destination directories which is executed every hour. There is no other mechanism involved other than the AWS DataSync server.

- 
- `pbaws-datasync create -sdf <source_dirs_file>`: DataSync and subsequent import is done using a `cron` job and the AWS DataSync server. A user `cron` job is executed every 30 minutes. The user `cron` job is created on the host where the `pbaws-datasync create` command is run. The `cron` job uses list of directories listed in the file `<source_dirs_file>` that should be synchronized to AWS. In each subdirectory, the `cron` job looks for a `.transferdone` file; this is a signal that the directory is ready for synchronization to AWS. The `cron` job submits an individual DataSync task for each directory that has a `.transferdone` file and also monitors the transfer. When the transfer is completed, the `cron` job automatically submits an `import-dataset` job for that data.

### DataSync log files

- The `cloud-init-output`, `smrtlink`, and `cromwell` logs are available in the CloudWatch, `loggroup: /aws/<stack-name>`. For DataSync performed using the AWS DataSync infrastructure, the logs are available in `loggroup /aws/<stack_name>/datasync`.
- `cloud-init-output.log`: The very end of the log file has details about installation issues/failures in case `pbawstools create` failed when trying to install SMRT Link software on the EC2 instance.

### Configuring the AWS SMRT Link instance to send email using AWS SES

1. Verify the sender's email address using this [page](#).
2. Change the email account from `Sandbox` to `Production` by clicking the `Edit your account details` button on this [page](#).
3. Create the SMTP user using the `Create My SMTP credentials` button on this [page](#).
4. You must use the `smtp-user` and password used in the `-acfp` arguments `SmrtLinkMailUser` and `SmrtLinkMailPassword` used with the `pbawstools create` or `pbawstools update` commands. By default, `SmrtLinkMailHost` is set as `email-smtp.us-west-2.amazonaws.com` and port 587. Change the `SmrtlinkMailHost` name to the one corresponding to your stack region using the SMTP Endpoints section on this [page](#).
5. If sending email from SMRT Link does not work, first test sending from the headnode EC2 terminal using instructions on this [page](#).
6. If the email is not received, contact AWS support and make sure that your account does not have any other restrictions for sending email.
7. If the email is still not received, check the spam filter/folder and make sure the email from AWS is not quarantined there.

---

## Appendix A: Creating an AWS Amazon account

The following instructions assume that you don't already have an AWS Amazon account.

1. Navigate to the AWS sign up page at <https://aws.amazon.com/>.
2. Click **Create an AWS account**.
3. Enter an email address, password, and AWS account name, then click **Continue**.
4. Enter contact and payment information.
5. Confirm your identity.
6. Select **Basic Plan**, then click **Free**.

**Note:** We **strongly** recommend subscribing to an AWS support plan – see [here](#) for details. Note that for production-level workload, Amazon suggests **Business**-level support.

### Additional information:

- **Amazon Elastic Compute Cloud (Amazon EC2):** See [here](#) for documentation.
- **Create and activate a new Amazon AWS EC2 account:** See [here](#).



## Appendix B: PacBio command-line cloud utilities

This section describes the command-line tools included with SMRT Link v11.0 for installing and working with SMRT Link Cloud.

- The command-line tools are located in the `$SMRT_ROOT/smrtlink/smrtcnds/bin` subdirectory.

**pbawtools** The `pbawstools` tool is used to work with AWS SMRT Link resources and services. The tool lets you create, stop, start, update and delete SMRT Link resources from your AWS instance.

### Usage

```
pbawstools [-h] [--version] [--log-file LOG_FILE]
           [--log-level {DEBUG,INFO,WARNING,ERROR,CRITICAL} | --debug | --quiet | -v]
           {delete,create,update,start,stop,url}
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--version</code>	Displays program version number and exits.
<code>--log-file</code>	Writes the log to a file. (Default = <code>stdout</code> )
<code>--log-level</code>	Specifies the log level; values are <code>[DEBUG, INFO, WARNING, ERROR, CRITICAL]</code> . (Default = <code>INFO</code> )
<code>--debug</code>	Alias for setting the log level to <code>DEBUG</code> . (Default = <code>False</code> )
<code>--quiet</code>	Alias for setting the log level to <code>CRITICAL</code> to suppress output. (Default = <code>False</code> )
<code>-v, --verbose</code>	Specifies the verbosity level. (Default = <code>None</code> )

**create** Command: Create AWS SMRT Link resources and services.

```
pbawstools create [-h] [--template_file TEMPLATE_FILE]
                  [--aws_keypair_name KEYPAIR_NAME]
                  [--vpcid VPC_ID | --create_vpc]
                  [--azones {a b,a c,b c,a b c}]
                  [--subnets_id SUBNETS_ID]
                  [--priv_subnets_id PRIV_SUBNETS_ID]
                  [--stack_name STACK_NAME]
                  [--aws_keyId AWS_KEYID]
                  [--aws_secretKey AWS_SECRETKEY]
                  [--aws_region AWS_REGION] --ssh_pem_key SSH_PEM_KEY
                  [--aws_cf_parameters AWS_MISC_PARAMS | --aws_cf_parameters_file
AWS_MISC_PARAMS_FILE]
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--template_file, -tf</code>	Specifies the absolute path of the cloud formation template json file.
<code>--aws_keypair_name, -akp</code>	Specifies the AWS Key-pair name.
<code>--vpcid, -vid</code>	Specifies the ID of the existing Virtual Private Cloud (VPC). See <a href="https://aws.amazon.com/vpc/">https://aws.amazon.com/vpc/</a> for details.

Options	Description
--azones, -azs	Specifies the availability zones as space-separated strings within quotes; dependent on --create_vpc. (Default = "a b c")
--subnetsid, -sbnid	Specifies two subnet IDs in the given Virtual Private Cloud as space-separated strings within quotes. Required only if --vpcid is present. <b>Example:</b> subnet1_id subnet2_id
--priv_subnets_id, -psbnid	Specifies two private subnet IDs in the given Virtual Private Cloud as space-separated strings within quotes. Dependent on --vpcid <b>Example:</b> subnet1_id subnet2_id"
--create_vpc, -cvpc	Creates a new AWS Virtual Private Cloud and subnets for this stack, mutually exclusive with --vpcid.
--stack_name, -sn	Specifies the stack name to create to the SMRT Link services. Note: The stack name must satisfy the following regular expression pattern: [a-zA-Z] [-a-zA-Z0-9]*
--aws_keyId, -akId	Specifies the AWS Key ID.
--aws_secretKey, -ask	Specifies the AWS Secret Key.
--aws_region, -arg	Specifies the AWS Region.
--ssh_pem_key, -spk	Specifies the absolute path of the ssh pem key file.

Options	Description
<pre>--aws_cf_parameters, -acfp</pre>	<p>Specifies optional AWS parameters as key-value pairs to set custom values. Values are:</p> <ul style="list-style-type: none"> <li>- InstanceType: The type of EC2 instance to be used for the head node. (Default = r5.xlarge)</li> <li>- SmrtlinkDocker: AWS batch job docker container location. (Default = quay.io/pacificbiosciences/sl-aws-batch:latest)</li> <li>- SmrtlinkNproc: nproc flag to be used in the SMRT Link configuration. (Default = 8)</li> <li>- SmrtlinkNworkers: Number of non-blocking jobs that can run simultaneously. (Default = 10)</li> <li>- SmrtLinkMailHost: SMTP email server to use for sending email from SMRT Link. (Default = email-smtp.us-west-2.amazonaws.com)</li> <li>- SmrtLinkMailPort: Email server port. (Default = 587)</li> <li>- SmrtLinkMailUser: SMTP Email server.</li> <li>- SmrtLinkMailPassword: SMTP email password.</li> <li>- Smrtlinkmaxchunks: Maximum number of chunks allowed in a SMRT Link task. (Default = 96)</li> <li>- BatchJobMemory: Memory requested per core to be used in case it is not specified in the job/workflow. (Default = 4GB)</li> <li>- SmrtLinkSoftwareLink: http location to download the SMRT Link tarball. This should be a publicly downloadable location. If using S3, that s3-file should be set to public-read.</li> <li>- SoftwareLinkPassword: The password for the SMRT Link software if there is one.</li> <li>- SSHLocation: The IP address range that has SSH access to the EC2 instances. (Default = 0.0.0.0/0)</li> <li>- HTTPLocation: The IP address range that has HTTP access to the EC2 instances. (Default = 0.0.0.0/0)</li> <li>- OnDemandCEMinvCpus: The minimum number of CPUs available all the time by the AWS Batch on-demand backend. (Default = 0)</li> <li>- OnDemandCEMaxvCpus: The maximum number of CPUs used by the AWS Batch on-demand backend. (Default = 5000)</li> <li>- SpotCEMinvCpus: The minimum number of CPUs to be available all the time by the AWS Batch spot backend. (Default = 0)</li> <li>- SpotCEMaxvCpus: The maximum number of CPUs that will be used by the AWS Batch spot backend. (Default = 5000)</li> <li>- PacBioInternal: AWS SMRT Link can access the PacBio Update and Event server if set to true. (Default = false)</li> <li>- SpotBidPercentage: How much to bid for the spot instances as a percentage of the cost of on-demand instances. (Default = 100)</li> <li>- FileSystemsList: Specifies other EFS or NFS-mountable file systems to mount to this SMRT Link instance. This is needed when data needs to be shared between different user account and/or instances. The file system must be specified using the "filesystemID:mount-directory" format.</li> <li>- BatchInstanceTypes: Comma-separated string of instance types to be used on the Batch backend. (Default = optimal)</li> <li>- BatchAllocationStrategy: Allocation strategy to be used by AWS Batch on-demand backend, one of BEST_FIT or BEST_FIT_PROGRESSIVE. (Default = BEST_FIT)</li> <li>- UsePacBioTestedAmi: Use the AMI tested by PacBio for the SMRT Link server if true. If false, the AMI recommended by AWS is used. (Default = true)</li> </ul> <p><b>Example:</b> { 'BatchJobMemory': '4', 'SmrtlinkNproc': '4', 'SSHLocation': '0.0.0.0/0', 'HTTPLocation': '0.0.0.0/0', 'SmrtLinkMaxchunks': '25', 'FileSystemsList': 'filesystemID1:/my-data1,fsID2:/my-data2' }</p>

Options	Description
<code>--aws_cf_parameters_file -acfpf</code>	Specifies a file name containing optional AWS parameters as key-value pairs to set custom values. (For parameter values, see <code>--aws_cf_parameters</code> above.) Mutually exclusive with <code>--awscfp</code> .

`delete` Command: Delete AWS SMRT Link resources and services.

```
pbawstools delete [-h] --stack_name STACK_NAME [--aws_keyId AWS_KEYID]
                  [--aws_secretKey AWS_SECRETKEY]
                  [--aws_region AWS_REGION]
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--stack_name, -sn</code>	Specifies the stack name to delete to the SMRT Link services.
<code>--aws_keyId, -akId</code>	Specifies the AWS Key ID.
<code>--aws_secretKey, -ask</code>	Specifies the AWS Secret Key.
<code>--aws_region, -arg</code>	Specifies the AWS Region.

`update` Command: Update AWS SMRT Link resources and services.

```
pbawstools update [-h] --stack_name STACK_NAME [--aws_keyId AWS_KEYID]
                  [--aws_secretKey AWS_SECRETKEY]
                  [--aws_region AWS_REGION] --ssh_pem_key SSH_PEM_KEY
                  [--aws_cf_parameters AWS_MISC_PARAMS | --aws_cf_parameters_file
                  AWS_MISC_PARAMS_FILE]
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--stack_name, -sn</code>	Specifies the stack name to update to the SMRT Link services.
<code>--aws_keyId, -akId</code>	Specifies the AWS Key ID.
<code>--aws_secretKey, -ask</code>	Specifies the AWS Secret Key.
<code>--aws_region, -arg</code>	Specifies the AWS Region.
<code>--ssh_pem_key, -spk</code>	Specifies the absolute path of the <code>ssh pem</code> key file.
<code>--aws_cf_parameters, -acfp</code>	Specifies optional AWS parameters as key-value pairs to set custom values. Values are: <ul style="list-style-type: none"> <li>- <code>SmrtLinkSoftwareLink</code>: http location to download the SMRT Link tarball.</li> <li>- <code>SmrtlinkNproc</code>: nproc flag to be used in the SMRT Link configuration. (Default = 8)</li> <li>- <code>Smrtlinkmaxchunks</code>: Maximum number of chunks allowed in a SMRT Link task. (Default = 96)</li> <li>- <code>SmrtlinkNworkers</code>: Number of non-blocking jobs that can run simultaneously. (Default = 10)</li> <li>- <code>SmrtLinkMailHost</code>: SMTP Email server to use for sending email from SMRT Link. (Default = <code>email-smtp.us-west-2.amazonaws.com</code>)</li> <li>- <code>SmrtLinkMailPort</code>: Email server port. (Default = 587)</li> <li>- <code>SmrtLinkMailUser</code>: SMTP Email server.</li> <li>- <code>SmrtLinkMailPassword</code>: SMTP email password.</li> </ul>

Options	Description
<code>--aws_cf_parameters_file -acfpf</code>	Specifies a file name containing optional AWS parameters as key value pairs to set custom values. (For parameter values, see <code>--aws_cf_parameters</code> above.)

`start` Command: Start AWS SMRT Link services using existing resources.

```
pbawstools start [-h] --stack_name STACK_NAME [--aws_keyId AWS_KEYID]
                [--aws_secretKey AWS_SECRETKEY]
                [--aws_region AWS_REGION] --ssh_pem_key SSH_PEM_KEY
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--stack_name, -sn</code>	Specifies the stack name to start to the SMRT Link services.
<code>--aws_keyId, -akId</code>	Specifies the AWS Key ID.
<code>--aws_secretKey, -ask</code>	Specifies the AWS Secret Key.
<code>--aws_region, -arg</code>	Specifies the AWS Region.
<code>--ssh_pem_key, -spk</code>	Specifies the absolute path of the <code>ssh pem</code> key file.

`stop` Command: Stop AWS SMRT Link services.

```
pbawstools stop [-h] --stack_name STACK_NAME [--aws_keyId AWS_KEYID]
                [--aws_secretKey AWS_SECRETKEY]
                [--aws_region AWS_REGION] --ssh_pem_key SSH_PEM_KEY
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--stack_name, -sn</code>	Specifies the stack name to stop to the SMRT Link services.
<code>--aws_keyId, -akId</code>	Specifies the AWS Key ID.
<code>--aws_secretKey, -ask</code>	Specifies the AWS Secret Key.
<code>--aws_region, -arg</code>	Specifies the AWS Region.
<code>--ssh_pem_key, -spk</code>	Specifies the absolute path of the <code>ssh pem</code> key file.

`url` Command: Obtain the AWS SMRT Link services url.

```
pbawstools url [-h] --stack_name STACK_NAME [--aws_keyId AWS_KEYID]
               [--aws_secretKey AWS_SECRETKEY]
               [--aws_region AWS_REGION]
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--stack_name, -sn</code>	Specifies the name of the stack whose URL to obtain to the SMRT Link services.
<code>--aws_keyId, -akId</code>	Specifies the AWS Key ID.
<code>--aws_secretKey, -ask</code>	Specifies the AWS Secret Key.

Options	Description
<code>--aws_region, -arg</code>	Specifies the AWS Region.

**pbaws-datasync** The `pbaws-datasync` tool is used to create and delete DataSync agents, locations, and tasks.

### Usage

```
pbaws-datasync [-h] [--version] [--log-file LOG_FILE]
                [--log-level {DEBUG,INFO,WARNING,ERROR,CRITICAL} | --debug | --quiet | -
v]
                {delete,create}
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--version</code>	Displays program version number and exits.
<code>--log-file</code>	Writes the log to a file. (Default = stdout)
<code>--log-level</code>	Specifies the log level; values are [DEBUG, INFO, WARNING, ERROR, CRITICAL]. (Default = INFO)
<code>--debug</code>	Alias for setting the log level to DEBUG. (Default = False)
<code>--quiet</code>	Alias for setting the log level to CRITICAL to suppress output. (Default = False)
<code>-v, --verbose</code>	Specifies the verbosity level. (Default = None)

**create Command:** Create a DataSync agent, locations and tasks.

```
pbaws-datasync create [-h] --agentIP AIP
                    [--src_dir DATA_TO_AWS_DIR | --src_dirs_file SRC_DIRS_FILE)
                    [--aws_dest_dir AWS_DEST_DIR]
                    [--aws_src_dir AWS_DIR]
                    [--dest_dir DATA_FROM_AWS_DIR]
                    [--efs_arn EFS_ARN] [--subnet_arn SUBNET_ARN]
                    [--sg_arn SG_ARN] [--ssh_pem_key SSH_PEM_KEY]
                    [--smrtlink_user SMRTLINK_USER]
                    [--smrtlink_password SMRTLINK_PASSWORD]
                    [--smrtlink_support_email SMRTLINK_SUPPORT_EMAIL]
                    [--stack_name STACK_NAME] [--aws_keyId AWS_KEYID]
                    [--aws_secretKey AWS_SECRETKEY]
                    [--aws_region AWS_REGION]
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--agentIP, -aip</code>	Specifies the IP address of the on-premise agent host.
<code>--src_dir, -sd</code>	Specifies the source directory containing the files to be transferred to AWS.
<code>--src_dirs_file, -sdf</code>	Specifies the absolute path of the file that contains the list of directories to be transferred to AWS, as well as the Data Sets to be imported to SMRT Link
<code>--aws_dest_dir, -add</code>	Specifies the AWS directory containing the files to be transferred from AWS.

Options	Description
<code>--dest_dir, -dd</code>	Specifies the AWS destination directory that contains the files that were transferred from AWS.
<code>--subnet_arn, -sna</code>	Specifies the subnet Amazon Resource Name (ARN) in AWS of the target Amazon Elastic File System (EFS). See <a href="#">here</a> and <a href="#">here</a> for details.
<code>--efs_arn, -ea</code>	Specifies the EFS ARN in AWS that hosts the synchronization directory.
<code>--sg_arn, -sga</code>	Specifies the Security Group ARN in AWS of the target EFS.
<code>--ssh_pem_key, -spk</code>	Specifies the absolute path of the <code>ssh pem</code> key file.
<code>--smrtlink_user, -slu</code>	Specifies the SMRT Link user name of the user using the SMRT Link Cloud instance.
<code>--smrtlink_password, -slp</code>	Specifies the SMRT Link password of the user using the SMRT Link Cloud instance.
<code>--smrtlink_support_email -sle</code>	Specifies the SMRT Link support email details to notify of import/transfer errors. <b>Example:</b> <code>{'server':'mail-server.xxx.com','to':'to_email_address','from':'frpm_email_address' }</code>
<code>--stack_name, -sn</code>	Specifies the cloud formation stack to which this DataSync agent serves. (This is the same <code>stack_name</code> specified in <code>pbawstools create</code> .)
<code>--aws_keyId, -akId</code>	Specifies the AWS Key ID.
<code>--aws_secretKey, -ask</code>	Specifies the AWS Secret Key.
<code>--aws_region, -arg</code>	Specifies the AWS Region.

`delete` Command: Delete a DataSync agent, locations and tasks.

```
pbaws-datasync delete [-h] [--stack_name STACK_NAME]
                    [--aws_keyId AWS_KEYID]
                    [--aws_secretKey AWS_SECRETKEY]
                    [--aws_region AWS_REGION]
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--stack_name, -sn</code>	Specifies the cloud formation stack to which this DataSync agent serves. (This is the same <code>stack_name</code> specified in <code>pbawstools create</code> .)
<code>--aws_keyId, -akId</code>	Specifies the AWS Key ID.
<code>--aws_secretKey, -ask</code>	Specifies the AWS Secret Key.
<code>--aws_region, -arg</code>	Specifies the AWS Region.

**pbaws-efpsync** The `pbaws-efpsync` tool is used to synchronize data **to** and **from** AWS, and optionally locally.

### Usage

```
pbaws-efpsync [-h] [--version] [--log-file LOG_FILE]
              [--log-level {DEBUG,INFO,WARNING,ERROR,CRITICAL} | --debug | --quiet | -v]
              {to_aws,from_aws,local}
```

Options	Description
-h, --help	Displays help information and exits.
--version	Displays program version number and exits.
--log-file	Writes the log to a file. (Default = stdout)
--log-level	Specifies the log level; values are [DEBUG, INFO, WARNING, ERROR, CRITICAL]. (Default = INFO)
--debug	Alias for setting the log level to DEBUG. (Default = False)
--quiet	Alias for setting the log level to CRITICAL to suppress output. (Default = False)
-v, --verbose	Specifies the verbosity level. (Default = None)

`to_aws` Command: Synchronize data **from** the local source **to** the AWS destination.

```
pbaws-efpsync to_aws [-h]
                    [--stack_name STACK_NAME | --sync_host_dns SYNC_HOST]
                    [--ssh_pem_key SSH_PEM_KEY --source SRCDIR --dest]
                    DESTDIR [--rsync_options RSYNC_OPTIONS]
                    [--jobscount JOBS_COUNT]
                    [--fpart_options FP_OPTIONS] [--import-datasets]
                    [--smrtlink-user SMRTLINK_USER]
                    [--smrtlink-password SMRTLINK_PASSWORD]
                    [--block-for-import]
```

Options	Description
-h, --help	Displays help information and exits.
--stack_name, -sn	Specifies the AWS stack of the resources.
--sync_host_dns, -shd	Specifies the AWS EC2 host that will used for synchronizing data.
--source, -s	Specifies the source directory.
--dest, -d	Specifies the destination directory.
--rsync_options, -r	Specifies the <code>rsync</code> options. Defaults are: <ul style="list-style-type: none"> <li>- <code>-rpgDvLm --partial</code> for <code>to_aws/from_aws</code></li> <li>- <code>-rtDvLm --partial --chmod=D2775</code></li> <li>- <code>--chmod=F+r</code> for local</li> </ul>
--jobscount, -j	Specifies the number of parallel synchronization jobs.
--fpart_options, -f	Specifies <code>fpart</code> options. ( <code>fpart</code> sorts and packs files into partitions. See <a href="#">here</a> for details.)
--import-datasets	Assumes the AWS host is a SMRT Link server and attempts to import any XML Data Sets that were part of the transfer.
--smrtlink_user, -slu	Specifies the SMRT Link user name of the user using the SMRT Link Cloud instance.
--smrtlink_password, -slp	Specifies the SMRT Link password of the user using the SMRT Link Cloud instance.
--block-for-import	Specifies a wait for the import jobs to complete successfully.



`from_aws` Command: Synchronize data **from** the AWS source **to** the local destination.

```
pbaws-efpsync from_aws [-h]
                        [--stack_name STACK_NAME | --sync_host_dns SYNC_HOST]
                        [--ssh_pem_key SSH_PEM_KEY --source SRCDIR --dest DESTDIR]
                        [--rsync_options RSYNC_OPTIONS]
                        [--jobscount JOBS_COUNT]
                        [--fpart_options FP_OPTIONS]
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--stack_name, -sn</code>	Specifies the AWS stack of the resources.
<code>--sync_host_dns, -shd</code>	Specifies the AWS EC2 host that will used for synchronizing data.
<code>--ssh_pem_key, -spk</code>	Specifies the absolute path of the <code>ssh pem</code> key file.
<code>--source, -s</code>	Specifies the source directory.
<code>--dest, -d</code>	Specifies the destination directory.
<code>--rsync_options, -r</code>	Specifies the <code>rsync</code> options. Defaults are: <ul style="list-style-type: none"> <li>- <code>-rpgodvLm --partial</code> for <code>to_aws/from_aws</code></li> <li>- <code>-rtDvLm --partial --chmod=D2775</code></li> <li>- <code>--chmod=F+r</code> for local</li> </ul>
<code>--jobscount, -j</code>	Specifies the number of parallel synchronization jobs.
<code>--fpart_options, -f</code>	Specifies <code>fpart</code> options. ( <code>fpart</code> sorts and packs files into partitions. See <a href="#">here</a> for details.)

`local` Command: Synchronize data from a **local** source to a **local** destination.

```
pbaws-efpsync local [-h] --source SRCDIR --dest DESTDIR
                    [--rsync_options RSYNC_OPTIONS]
                    [--jobscount JOBS_COUNT]
                    [--fpart_options FP_OPTIONS]
```

Options	Description
<code>-h, --help</code>	Displays help information and exits.
<code>--source, -s</code>	Specifies the source directory.
<code>--dest, -d</code>	Specifies the destination directory.
<code>--rsync_options, -r</code>	Specifies the <code>rsync</code> options. Defaults are: <ul style="list-style-type: none"> <li>- <code>-rpgodvLm --partial</code> for <code>to_aws/from_aws</code></li> <li>- <code>-rtDvLm --partial --chmod=D2775</code></li> <li>- <code>--chmod=F+r</code> for local</li> </ul>
<code>--jobscount, -j</code>	Specifies the number of parallel synchronization jobs.
<code>--fpart_options, -f</code>	Specifies <code>fpart</code> options. ( <code>fpart</code> sorts and packs files into partitions. See <a href="#">here</a> for details.)

---

## Appendix C: AWS security

- For all security questions and concerns: See [here](#).
- Amazon compliance offerings: See [here](#).
- GDPR (General Data Protection Regulation)
  - All AWS services are GDPR-ready: See [here](#).
  - GDPR Center: See [here](#).
  - White paper on Navigating GDPR Compliance on AWS: See [here](#).
- AWS and the USA Cloud Act: See [here](#).
- AWS certifications: See [here](#).
- AWS services by compliance and certification: See [here](#).
- Requesting AWS compliance reports using Artifact: See [here](#).

## Appendix D: Frequently asked questions

- AWS questions**
- Q: Should you set up one AWS account for multiple users?**
- The general practice is to set up **one** AWS account for **multiple** users. However, SMRT Link Cloud will also work well with one account per user.
- Data transfer**
- Q: What are some other transfer mechanisms?**
- `pbaws-efpsync` (based on `fpsync`) multithreads the transfer easily, and also retries the transfer 3 times in case it fails.
  - `rsync` will also work well, but you must retry manually if it fails.
- Q: How do I monitor the progress of the DataSync?**
- See [“Monitoring DataSync progress” on page 12](#).
- Q: What file formats are included in the DataSync?**
- See [“File formats included in the DataSync” on page 10](#).
- Q: What mechanisms are used for the DataSync?**
- See [“Mechanisms used for the DataSync” on page 12](#).
- Q: Where do I find the log files for DataSync?**
- See [“DataSync log files” on page 13](#).
- Q: What are the technical specifications of the internal storage server used for the file transfer?**
- See the table on Page 3.
- Technical support**
- Q: Who is responsible for SMRT Link Cloud support?**
- PacBio is responsible for supporting SMRT Link Cloud. Any AWS-related questions and issues, such as how to create an account, billing, and so on, should be addressed to Amazon AWS Support.
- Q: How do I share the logs for troubleshooting with PacBio Technical Support?**
- There are two ways to do this:

1. Use CloudWatch to put the file into an S3 bucket and share the S3 bucket.
2. Use a customer-specific S3 bucket created by PacBio for troubleshooting purposes. PacBio Technical Support can provide appropriate permissions to use this option. For more information, see [here](#). Step 3 in the linked page lists the policies to follow for placing logs into a bucket owned by another account, such as PacBio.

**Q: When do I contact AWS support if `pbawstools create` runs into issues?**

- If there are any issues **before** line 5 is printed, it is likely to be some parameter issue; contact PacBio Technical Support.
- If the error is **after** line 5, but line 6 is not there yet, it is likely a VPC limits issue. Check the AWS **cloudformation** console for the stack results/error and contact AWS. (Lines 4 and 5 will be there **only** if `--create_vpc` is used.)
- If there are any issues **after** line 6, and line 7 is not there, contact PacBio Technical Support.
- Any issues **after** line 7 are very likely to be an AWS issue. Logon to cloudformation to see the issue and contact AWS if it is a resource/limits issue.  
If everything is created successfully, but `SmrtLinkServerInstance` resource creation failed on the cloudformation console, look in the CloudWatch Logs, log stream `/aws/<stack-name>,cloud-init-output.log` or send it to PacBio Technical Support. If there is **no** `cloud-init-output.log` file available, this is a resource issue; contact AWS support.

```
[INFO] 2020-10-06 21:28:42,288Z [pbawstools.sl_aws_pacbio_main_runner 160] Using
pbcommand v2.2.0
[INFO] 2020-10-06 21:28:42,288Z [pbawstools.sl_aws_pacbio_main_runner 163] completed
setting up logger with <function setup_log at 0x7f7b8ffd89e0>
[INFO] 2020-10-06 21:28:42,288Z [pbawstools.sl_aws_pacbio_main_runner 164] log opts
{'level': 20, 'file_name': None}
[INFO] 2020-10-06 21:28:53,377Z [pbawstools.sl_aws_createStack 282] Creating <stack-
name>-vpc
[INFO] 2020-10-06 21:28:55,509Z [pbawstools.sl_aws_createStack 285] ...waiting for
stack to be ready...
[INFO] 2020-10-06 21:32:27,360Z [pbawstools.sl_aws_createStack 282] Creating <stack-
name>
[INFO] 2020-10-06 21:32:28,222Z [pbawstools.sl_aws_createStack 285] ...waiting for
stack to be ready.
```

**Using SMRT Cloud**

**Q: Can data import failures be recovered?**

- Yes, data import failure may be recoverable by re-importing the Data Set.

**Q: How is SMRT Link Cloud exposed to the internet?**

- Access to SMRT Link Cloud is restricted to **only** the specific IPs and user accounts with secure access keys enabled by your SMRT Link Cloud administrator.

**Q: How to secure access to AWS?**

- We recommend restricting access to a SMRT Link Cloud instance to a specific IP range. This can be done during the account set up or using the AWS console. To do so, change the `slSecurityGroup` in bound rules.

**Q: How can I store and archive my data on the cloud?**

- See [“Storing and archiving data on the cloud” on page 7.](#)

---

**Q: What compute configurations are available in SMRT Link Cloud and what is the difference between them?**

- Separate configurations for both spot and on-demand instances are provided, each with various options for `maxchunks` and `nproc` (for a total CPU core count of 768). The spot instances are the default as they are **significantly cheaper**, but this may result in a longer wait for available instances.

**Q: Can I change SMRT Link Cloud compute resources to run analysis faster, or schedule it for time when AWS resources are cheaper?**

- To lower the cost of an analysis, use **spot instances**. We have not tested configurations other than those we provide. Note that requesting many large nodes may be **significantly more expensive** and may also increase the wait time for individual jobs.

**Q: What are the pros and cons of using on-demand versus spot instances?**

- Spot instances are **cheaper**, but may take longer to obtain. Their main restriction is the number of simultaneous instances that can be used. Spot instances could be terminated when the job is running. For most light analyses, spot instances will probably be the best choice.
- On-demand instances provide maximum throughput if you have many jobs to run simultaneously. They are usually **more expensive** than spot instances.

**Q: Is there LDAP integration for SMRT Link Cloud? Can I use my network login?**

- We recommend that the site administrator creates an individual account in WSO2 for **each** user for now.

**Q: How can I share analysis results and sequencing data with collaborators?**

To share a Data Set and/or analysis results, use the SMRT Link Export feature to create a zip file for export. Once the zip file is created, use one of the following AWS options for sharing data:

**Option 1: Sharing using S3 buckets.** The data to be shared can be organized into one or more S3 buckets by the owner. The owner can give permissions to accounts with which the data needs to be shared.

**Note:** The SMRT Link Cloud data shared using S3 buckets **must** be downloaded to a file system to be used as directories and files in SMRT Link or elsewhere, as S3 is only an object store.

**Sharing S3 buckets between accounts:**

- Cross-account access to objects in Amazon S3 buckets: See [here](#) for details.
- Bucket owner granting cross-account bucket permissions: See [here](#) for details.
- Bucket owner granting cross-account permission to objects it does not own: See [here](#) for details.

**Option 2: Sharing using EFS**

- Accessing an EFS file system across accounts using IAM authorization and EFS Access Points: See [here](#) for details.
- Mounting EFS file systems from another account or VPC. See [here](#) for details.

- 
- **Warning:** Symbolic linking does **not** work for EFS.

### Option 3: Sharing using AWS DataSync

- Setup a DataSync transfer between source EFS or S3 (owned by account A) to target EFS or S3 (owned by account B) to synchronize data one way or back and forth.

**Option 4: Sharing via FSx lustre** (See [here](#) for details.) This is a new file service similar to EFS, which can serve files from S3 buckets **without** explicit download (except the first time the file accessed) and write back to S3 when done.

- Data in FSx lustre can be shared using VPC peering: See [here](#) for details.

To make shared data available to SMRT Link, EFS sharing (**Option 2**) and/or sharing via FSx lustre (**Option 4**) are the most efficient way.

**Q: What is the idle running cost of the SMRT Link instance? What is the distributed computing backend (i.e. via Batch)?**

- See [“Stopping and restarting SMRT Link Cloud” on page 8](#).

**Q: Can the administrator password be changed or will that cause issues?**

- Access your SMRT Link Cloud instance's command line using `ssh` (See [“ssh access” on page 8](#) for details). Then enter the following command:

```
/pacbio-root/pacbio-software/smrtlink/admin/bin/set-wso2-creds --user admin --password <password>
```

**Q: How do I configure the AWS SMRT Link instance to send email using AWS SES?**

- See [“Configuring the AWS SMRT Link instance to send email using AWS SES” on page 13](#).

**Q: How and where do I find information about who is running what analysis and the cost associated with it?**

- See [“Monitoring SMRT Link Cloud usage” on page 8](#).

**Q: The message `You have new mail in /var/spool/mail/ec2-user` is printed when I logon to the head node EC2 instance. How do I stop this message?**

- The above message is from the `crons` run on the EC2 instance for monitoring by `cloudwatch-agent`. To suppress this message, add the line `unset MAILCHECK` to `~/ .bashrc` on the EC2 instance.
- `echo "unset MAILCHECK" >> ~/ .bashrc`

**Q. Can I SSH into the head node and run SMRT Tools commands there?**

- **Warning:** We have **not** tested this option and **discourage** the use of `ssh`.
- Tools that interact with the SMRT Link server and jobs directly, namely `pbservice`, `export-datasets`, and `export-job`, are fine to use, but we recommend keeping head node use to an absolute minimum.

**Q: What are the different VPC (Virtual Private Cloud) options available for AWS SMRT Link creation with `pbawstools create`?**

A VPC is **required** for AWS Batch and other AWS services.

- 
1. **Default VPC:** By default the stack is created on the default VPC. The default VPC must be for the region where the stack is created.
  2. **User-specified non-default preexisting VPC:** This requires a `vpc-id` and two `subnet-ids` entered on the command line.
  3. **(Recommended for production instances)** A new VPC created with the `--create_vpc` option and dedicated to SMRT Link. This option uses default classless inter-domain routing (CIDR) addresses. To use only a specific CIDR range, it is better to create a VPC using AWS console/cli and then use Option 2 above to create the stack.

VPCs listed in Options 1 and 2 may have many other instances or network operations. As the SMRT Link infrastructure will be on the same network, SMRT Link operations may be affected.

Ensure that the AWS account is within the limits of the VPC resources listed [here](#). Note that these are the **default** VPC limits - they can be increased by contacting AWS support.

### Q: What are the IAM permissions required to create SMRT Link instance on AWS?

- Following are broadly the set of IAM (Identity and Access Management, an AWS service) permissions required for creating and running a SMRT Link instance on AWS if administrator access **cannot** be given to an IAM user. Create a policy using this json and attach it to a group or IAM user.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:*",
        "batch:*",
        "cloudformation:*",
        "logs:*",
        "elasticloadbalancing:*",
        "autoscaling:*",
        "iam:*",
        "elasticfilesystem:*",
        "kms:*",
        "s3:*",
        "ecs:*",
        "datasync:*",
        "fsx:*",
        "tag:*",
        "resource-groups:*",
        "ssm:GetParameters",
        "sts:AssumeRole"
      ],
      "Resource": "*"
    }
  ]
}
```