TandemViz AWS Marketplace

User Guide

TandemViz Version 3.5.19

Dec 2025

tandemviz-support@tandemai.com

1. Scopes

This AWS Marketplace product automates the process of the provisioning of computational resources and configuration of TandemViz™ services in the AWS Cloud. This document covers the steps to launch TandemViz™ in AWS.

TandemViz™ is TandemAI's proprietary, web-based platform that provides our drug discovery clients with a user-friendly and collaborative graphical interface to our suite of cutting-edge AI and physics-based computational tools (called TandemOS). The TandemViz™ platform, which can be accessed from anywhere through a web browser, allows the seamless integration and visualization of experimental data alongside computational analyses.

For
more
information

about

TandemViz™

please

visit https://tandemai.com/platform/tandemviz.

1.1. Product disclosure

1.1.1. Customer Data

TandemViz™ collects and stores user details, such as names and email addresses, for authentication and notification purposes. This information is securely stored within the VPC deployment database and is never shared with third parties. The following fields are required for the CloudFormation deployment (as outlined in Section 3):

- AdminEmail: email address of the email. This email address is used for authentication, and then for multifactor authentication.
- Details of an SMTP account (NoReplyEmailSMTPHost, NoReplyEmailSMTPPort, NoReplyEmailAccount, NoReplyEmailPassword): these details are used by TandemViz to send notifications and multi-factor authentication code for the accounts.

1.1.2. Use of TandemAI's lambda layers

This product utilizes TandemAI's custom Lambda layers as part of the process to provision computational resources and configure TandemViz. These layers are hosted in TandemAI's S3 bucket and they can be found here:

- https://tandemaiserverless.s3.amazonaws.com/master/createuser_layer.zip

This lambda layer is used to create internal cluster accounts needed for TandemViz. These cluster accounts are created with random passwords and cannot be logged in from outside.

- https://tandemaiserverless.s3.amazonaws.com/master/lambda-layer-pcluster361.zip

This lambda layer is used to bring up a parallel cluster (https://aws.amazon.com/hpc/parallelcluster/).

2. PreConfigure

2.1. AWS account

It is recommended to run this solution under an administrator (admin) account since it requires your role to be able to launch a few different services, such as OpenSearch, RDS, EC2, and VPC. A base Identity and Access Management (IAM) policy with necessary permission to run TandemVPC is shown in Appendix A. It is recommended to deploy TandemVPC into a new AWS account because the permissions allow TandemVPC to create administrative roles, users, or groups.

To deploy TandemViz™, familiarity with AWS and Linux system administration is required. Please contact us if you need assistance.

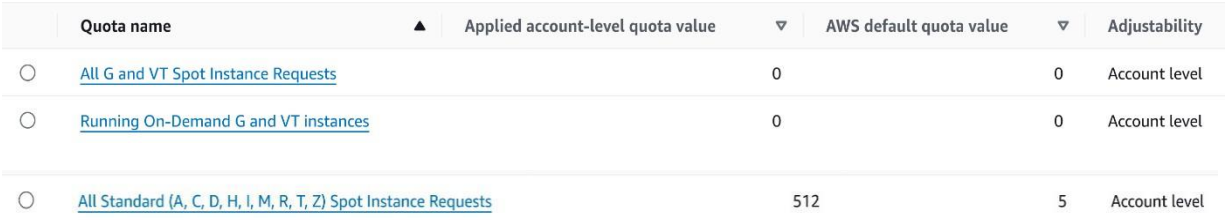
2.2. AWS License Manager

Please make sure you have your TandemViz™ license activated in AWS License Manager. Otherwise, you will not be able to create accounts. Please contact tandemvizsupport@tandemai.com if you need any help with the TandemViz™ license.

2.3. Increase your EC2 service quota for your AWS account

TandemVPC can make use of both ***on-demand*** and ***spot*** CPU and GPU instances for computation. Depending on your usage, you might need to increase your EC2 service quota for either on-demand or spot instances. You can decide to increase both spot and ondemand instances to utilize both.

These are the default quotas for a typical account in AWS East 1 region.



Quota name	Applied account-level quota value	AWS default quota value	Adjustability
All G and VT Spot Instance Requests	0	0	Account level
Running On-Demand G and VT instances	0	0	Account level
All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests	512	5	Account level

Note that these calculations are based on the cluster size we recommend. TandemVPC offers customizable resource configurations for system creation. Please contact us at TandemViz-Support@tandemai.com for any configuration queries.

Small Cluster Size:

- **All G and VT Spot Instance Requests**: increase to 280
- **All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests**: No increase
- **Running On-Demand G and VT Instances**: increase to 280

Medium Cluster Size

- **All G and VT Spot Instance Requests**: increase to 840
- **All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests**: increase to 1440
- **Running On-Demand G and VT Instances**: increase to 840

Large Cluster Size

- **All G and VT Spot Instance Request**: increase to 1680
- **All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests**: increase to 2880
- **Running On-Demand G and VT Instances**: increase to 1680

XLarge Cluster Size

- **All G and VT Spot Instance Request**: increase to 3360
- **All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests**: increase to 5760
- **Running On-Demand G and VT Instances**: increase to 3360

2.4. Increase your Elastic IPs (EIPs) quota for your AWS account

TandemVPC requires one elastic IPs per **a**vailability **z**one (AZ) for Network Address Translation (NAT) gateway and one elastic IP for the bastion node. Please increase your account's EIP quota to meet this minimum requirement.

2.5. An SMTP account

TandemViz™ requires an SMTP account for sending out OTP (one-time passcode) verification, notifications, etc. You need this SMTP account to log into TandemViz™.

Please contact us at TandemViz-Support@tandemai.com if you do not have an SMTP account. We can help you create one.

2.6. Using your domain name

A new AWS Certificate with AWS Certificate Manager will need to be created if you want to use your own domain name. Once the certificate is created, you can supply

the certificate Amazon Resource Name (ARN) to the CloudFormation stack in the next section.

Instructions for creating an AWS Certificate can be found in Appendix C. Further instructions about AWS certificates can be found here: https://docs.aws.amazon.com/acm/latest/userguide/gs-Appeacm-request-public.html.

3. Stacks

We provide two options to launch TandemViz™:

1. tandemvpc-easy
2. tandemvpc

While option 1 is easier to bring up, it is limited in terms of options. Option 2 exposes more parameters to customize your instance.

These two stacks create new IAM roles, the details of these roles and purposes are detailed in Appendix B.

The following steps cover how to launch from the AWS Web console. This can also be done using the command line.

3.1. TandemVPC-easy stack

3.1.1. Initiate a new stack

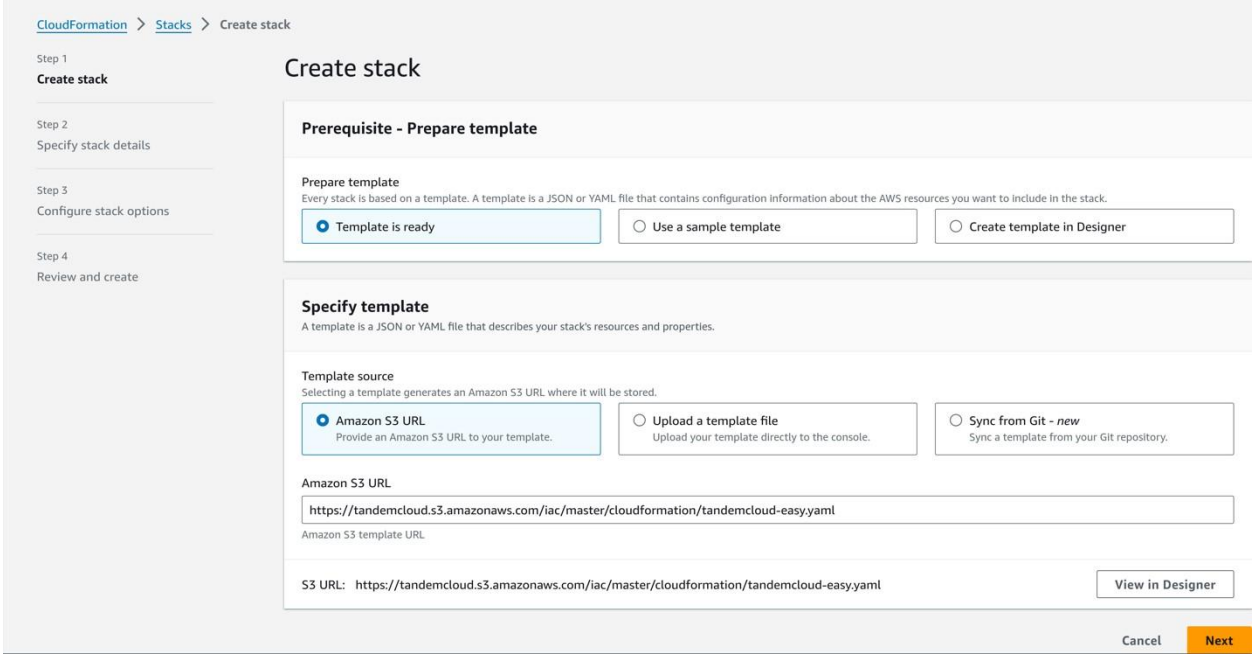
First, navigate to the Cloudformation console in the desired AWS account, then click **Create Stack**

For AWS China users, navigate to https://tandemcloud.s3.cn-northwest-

1.amazonaws.com.cn/iac/master/cloudformation/tandemvpc-easy.yaml

For AWS users outside of China, navigate to https://tandemcloud.s3.amazonaws.com/iac/master/cloudformation/tandemvpc-easy.yaml

If you start from AWS Marketplace, then this should be pre-filled.



3.1.2. Stack details

In this step, you need to fill in the necessary parameters

Stack name

Alphabets only and **lower case** and **less than 15 characters** in length.

Admin section

This parameter section covers all inputs related to admin account.

Email address for the admin account

Each account in TandemViz™ requires an account. And this is the email address associated with the very first admin account.

Password for the admin account

This is the initial password for the first admin account. Once the infrastructure is launched, you can log in with username **admin**, and this password.

You will be asked to change the password when logged in for the first time.

The size of your cluster

TandemVPC creates a Parallel Cluster to manage the computation jobs. This cluster uses Slurm as the job scheduler and uses multiple partitions (as listed below). The cluster will automatically spawn new instances (till a specified limit) to serve new jobs and delete unused instances.

- ***master***: this partition is only used for master jobs. Each master node is a t2.medium with 2 vCPUs and 4GBs of memory.
- ***cpu***: this partition is reserved for CPU only jobs. Each node has 32 vCPUs and 64 GBs of RAM. In the US, each node in CPU partition is of type c7i.8xlarge. In China, it is c6i.8xlarge.
- ***cpu-spot***: This partition functions similarly to the cpu partition but utilizes spot instances.
- ***lightgpu***: this partition is reserved for jobs requiring single GPU only. Each node in this partition belongs to type g4dn.2xlarge. Each node has 1 T4 Nvidia GPU and 8 vCPUs and 32 GBs of memory.
- ***lightgpu-spot***: This section functions similarly to the lightgpu partition, but utilizes spot instances.

The following partitions consist of both CPU and GPU compute resources.

- ***debug***: this partition is for debugging purposes. Each node in CPU compute resource has 8 vCPUs and 16 GB of RAM (c6i.2xlarge in China

and c7i.2xlarge elsewhere). The GPU compute resource part consists of g4dn.xlarge nodes.

- ***project***: this is the biggest partition in the cluster, it is for performing the most compute-intensive applications such as free-energy perturbation (FEP). Each CPU compute resource contains 16 vCPUs and 32 GBs of RAM (c7i.4xlarge in US and c6i.4xlarge in China). In China and California region, each GPU compute

  node is of type g4dn.12xlarge consisting of 4 Nvidia T4 GPUs, 48 vGPU and 192 GBs of RAM. In the other US regions, each GPU compute node is of type g6.12xlarge consisting of 4 Nvidia L4 GPUs, 48 vCPUs and 192 GBs of memory.
- ***project-spot***: this partition is the same as the project partition, but it uses spot instances.

The TandemVPC-easy stack has 4 configurations for the cluster with the following setups. Make sure you increase your quota accordingly in your AWS account

Small:

CpuQueueNodesMax: 10

LightGpuQueueNodesMax: 5

DebugQueueGPUNodeMax: 1

DebugQueueCPUNodesMax: 2

ProjectQueueCpuNodesMax: 10

ProjectQueueGpuNodesMax:5

Medium:

CpuQueueNodesMax: 30

LightGpuQueueNodesMax: 15

DebugQueueGPUNodeMax: 2

DebugQueueCPUNodesMax: 4

ProjectQueueCpuNodesMax: 30

ProjectQueueGpuNodesMax: 15

Large:

CpuQueueNodesMax: 60

LightGpuQueueNodesMax: 30

DebugQueueGPUNodeMax: 5

DebugQueueCPUNodesMax: 10

ProjectQueueCpuNodesMax: 60

ProjectQueueGpuNodesMax: 30

XLarge:

CpuQueueNodesMax: 120

LightGpuQueueNodesMax: 60

DebugQueueGPUNodeMax: 5

DebugQueueCPUNodesMax: 10

ProjectQueueCpuNodesMax: 120

ProjectQueueGpuNodesMax: 60

Whether to create a cluster in this stack

Whether to create a cluster in this stack. This option is used when upgrading the stack version.

Enable AD User/Group Mapping

Whether to enable AD User/Group mapping. Choose "yes" to connect the Viz and Cluster to your own AD, "no" to use default user management. When enabling AD User/Group mapping, you should do the following steps to make it effective to Viz and Cluster.

1. Create user "tandemviz" and "ReadOnlyUser" in your AD

2. Stop viz services on Backend node

```
/usr/local/bin/ansible-playbook --connection=local /opt/tandemai/ansible/ami_stopvizservices.yml
```

3. Update sssd configure for Backend node

3.1 get original uid of "tandemviz"

```
getent passwd tandemviz
```

3.2 update /etc/sssd/sssd.conf configure to your own AD

3.3 make it effective:

```
systemctl restart sssd
```

3.4 clean sssd cache:

```
sss_cache -E
```

3.5 check if the user "tandemviz" get new uid

```
getent passwd tandemviz
```

4. Update directory/file permission on Backend node

```
ls -l /nfs/home
chmod +rx /opt/tandemai
chown -R tandemviz /software/aws/
chown -R tandemviz /var/log/tandemai
chown -R tandemviz /nfs/projects/viz
chown tandemviz /nfs/projects/
```

chown -R tandemviz /nfs/tandemai/

chown -R tandemviz /nfs/home/tandemviz/

5. Update ffdb.sh on Backend node

vi /software/aws/ffdb/ffdb.sh

update "source /software/miniforge3/bin/activate apps_py38" to "source /software/miniforge3/bin/activate ta_force"

add "# load module" before "# init database"

load module

module load tabase/py38

module load taworkflow/v20_py38

```
# cp files
/bin/cp -f /software/aws/ffdb/$projectname/ffengine.cfg Force
/bin/cp -f /software/aws/ffdb/$projectname/alembic.ini Force

# load env params
source /etc/profile
source /software/miniforge3/bin/activate ta_force

# evn params setting
export PYTHONPATH=/software_common/ta_ffengine_latest:$PYTHONPATH
export FFENGINE_PKG_PATH=/software_common/ta_ffengine_latest/ffengine
export FFENGINE_HOME=$projectpath/$projectname/Force

# load module
module load tabase/py38
module load taworkflow/v20_py38

# init database
/software_common/ta_ffengine_latest/ffengine/bin/ffde initdb -a $projectpath/$projectname/Force/alembic.ini

# update the owner and permission
cd "$projectpath/$projectname"
#/bin/chown -R tandemviz:$projectname Force*
#/bin/chmod -R g+w Force_Data
#/bin/chmod -R g+w Force/alembic.ini
sudo /usr/bin/chown -R tandemviz:$projectname Force*
```

6. Update sssd configure for HeadNode

6.1 get original uid of "tandemviz"

getent passwd tandemviz

6.2 update /etc/sssd/sssd.conf configure to your own AD

6.3 make it effective:

```
systemctl restart sssd
```

6.4 clean sssd cache:

```
sss_cache -E
```

6.5 check if the user "tandemviz" get new uid

```
getent passwd tandemviz
```

7. Recreate slurm user on HeadNode

```
sacctmgr delete user tandemviz -i
sacctmgr create user tandemviz account=pcdefault admin=Administrator -i
```

8. Start viz services on Backend node

```
/usr/local/bin/ansible-playbook --connection=local /opt/tandemai/ansible/ami_startvizservices.yml
```

9. Project Creation:

9.1 Create AD project groups that correspond to Viz projects.

– Add the "tandemviz" account to these groups to grant access.

9.2 Viz operations:

– Create projects in Viz that map to these AD groups.

10. User Creation :

10.1 AD operations:

– Create AD user accounts for all users who need access to Viz.

– Assign users to the appropriate AD project groups.

10.2 Viz operations:

– Create user accounts in Viz that correspond to these AD users.

– Add users to the appropriate Viz projects based on their AD group memberships.

Bastion Host Settings

SSH location allowed to connect to bastion node.

Please set CIDR to x.x.x.x/32 to allow one specific IP address access, 0.0.0.0/0 to allow all IP addresses access, or another CIDR range.

VPC Networking Settings section

This section contains parameters for setting up the Virtual Private Cloud (VPC).

VPCId

The ID of your existing VPC. If you leave this parameter empty, a new VPC will be created.

InternetGatewayId

The ID of your existing Internet Gateway. If you leave this parameter empty, a new VPC will be created.

AZList

The list of availability zones (AZs) that this setup will use. ***Only the first two AZs will be used***.

VpcCIDR

The Classless Inter-Domain Routing (CIDR) of the VPC. This value is only used for new VPC.

CIDR First Public Network

The Classless Inter-Domain Routing (CIDR) of the first public network.

CIDR First Private Network

The Classless Inter-Domain Routing (CIDR) of the first private network.

CIDR Second Public Network

The Classless Inter-Domain Routing (CIDR) of the second public network.

CIDR Second Private Network

The Classless Inter-Domain Routing (CIDR) of the second private network.

Communications section

This section contains parameters of a no-reply SMTP email account used by TandemViz™ for sending out emails such as OTP or notifications.

SMTP Host

The host of the account. The default value is **smtp.office365.com**. Any other SMTP server works.

SMTP port

The port for connecting to the server. The default value is **587**.

Email account

The account for this no-reply email.

Email password

The password for this no-reply email.

FROM email address

The FROM email address for the non-reply email.

Others section

OpensearchServiceRoleExists

This lets the stack know whether an existing service role exists for Opensearch. To find out, go to IAM –> Roles –> search for AWSServiceRoleForAmazonOpenSearchService.

If this role does not exist, this stack will create a new role. This is needed for Opensearch Service, which is used by TandemViz™.

VizCertARN

The ARN of the Certificate created in previous step. If this ARN is empty, this stack will automatically create a self–signed certificate and the URL of this new instance of TandemViz™ is the default public domain name of the front–end node.

Opensearch Service Role Exists

Indicate if a service role for Opensearch is pre–existing within IAM. If absent, selecting "no" prompts the stack to create the necessary role for Opensearch Service, integral to TandemViz™ operations.

OpensearchVolumeSize

The size of the OpenSearch volume.

MPS3BucketName

Marketplace S3 Buket Name of CloudFormation Template. It's filled automatically by AWS.

MPS3BucketRegion

Marketplace S3 Buket Region of CloudFormation Template. It's filled automatically by AWS.

MPS3KeyPrefix

Marketplace S3 Buket Prefix of CloudFormation Template. It's filled automatically by AWS.

TandemVizAMI

This is the alias of the Marketplace AMI that will be deployed as part of this stack. It's filled automatically by AWS.

3.2. TandemVPC stack

This stack offers more control compared to the TandemVPC-easy stack, providing advanced customization options for users requiring advanced management over their cloud infrastructure setup for TandemViz™ services.

3.2.1. Initiate a new stack

First, navigate to the Cloudformation console in the desired AWS account, then click **Create Stack**

For AWS China users, navigate to https://tandemcloud.s3.cn-northwest-1.amazonaws.com.cn/iac/master/cloudformation/tandemvpc.yaml

For AWS users outside of China, navigate to https://tandemcloud.s3.amazonaws.com/iac/master/cloudformation/tandemvpc.yaml If you start from AWS Marketplace, then this should be pre-filled.

3.2.2. Stack details

Stack name

Alphabets only and **lower case**

VPC Networking Settings

VPCId

The ID of your existing VPC. If you leave this parameter empty, a new VPC will be created.

VpcCIDR

The Classless Inter-Domain Routing (CIDR) IP range within the VPC.

InternetGatewayId

The ID of your existing Internet Gateway. If you leave this parameter empty, a new VPC will be created.

VpcCIDR

The Classless Inter-Domain Routing (CIDR) of the VPC. This value is only used for new VPC.

CIDR First Public Network

The Classless Inter-Domain Routing (CIDR) of the first public network.

CIDR First Private Network

The Classless Inter-Domain Routing (CIDR) of the first private network.

CIDR Second Public Network

The Classless Inter-Domain Routing (CIDR) of the second public network.

CIDR Second Private Network

The Classless Inter-Domain Routing (CIDR) of the second private network.

Network File System Settings

BackupPolicy

Defines the backup policy for your file system. Options include enabling or disabling backups, which can be crucial for data recovery and compliance.

ThroughputMode

The throughput mode determines how the file system handles data transfer rates. "Bursting" allows for temporary spikes in data transfer beyond the baseline rate,

"Provisioned" lets you specify a fixed throughput rate, and "Elastic" adjusts throughput based on stored data volume.

PerformanceMode

Selects the performance mode for the file system. "GeneralPurpose" is suitable for most use cases, while "MaxIO" is optimized for large-scale, high-performance computing tasks.

Bastion Host Settings

BastionSSHLocation

Specifies the IP address range that is allowed to SSH into the Bastion host. This should be set to a secure range to prevent unauthorized access. The default value is 0.0.0.0. ***Database Cluster Configuration***

DBMinCapacity & DBMaxCapacity

These parameters define the scaling capacity of your database cluster, ensuring that it can adjust resources based on load while controlling costs.

Cluster configs

These parameters allow you to specify the types and sizes of instances and volumes for your computational cluster's head nodes and compute nodes, tailoring the cluster's performance and capacity to your needs. For detailed guidance on selecting the appropriate node numbers or GPU/CPU types, please refer to this section in the TandemVPC-easy stack documentation.

Viz Configs

VizCertARN

The ARN of the Certificate created in previous step. If this ARN is empty, this stack will automatically create a self-signed certificate and the URL of this new instance of TandemViz™ is the default public domain name of the front-end node.

OpensearchVolumeSize

The size of the OpenSearch volume.

Admin Email

Email address for the admin account.

Admin Password

Password for the admin account.

Opensearch Service Role Exists

Indicate if a service role for Opensearch is pre-existing within IAM. If absent, selecting "no" prompts the stack to create the necessary role for Opensearch Service, integral to TandemViz™ operations.

VizWebAccessLocation

Please set CIDR to x.x.x.x/32 to allow one specific IP address access, 0.0.0.0/0 to allow all IP addresses access, or another CIDR range

Enable AD User/Group Mapping

Whether to enable AD User/Group mapping. Choose "yes" to connect the Viz and Cluster to your own AD, "no" to use default user management. When enabling AD User/Group mapping, you should do the following steps to make it effective to Viz and Cluster.

11. Create user "tandemviz" and "ReadOnlyUser" in your AD

12. Stop viz services on Backend node

/usr/local/bin/ansible-playbook --connection=local /opt/tandemai/ansible/ami_stopvizservices.yml

13. Update sssd configure for Backend node

13.1 get original uid of "tandemviz"

getent passwd tandemviz

13.2 update /etc/sssd/sssd.conf configure to your own AD

13.3 make it effective:

systemctl restart sssd

13.4 clean sssd cache:

sss_cache -E

13.5 check if the user "tandemviz" get new uid

getent passwd tandemviz

14. Update directory/file permission on Backend node

ls -l /nfs/home

chmod +rx /opt/tandemai

chown -R tandemviz /software/aws/

chown -R tandemviz /var/log/tandemai

chown -R tandemviz /nfs/projects/viz

chown tandemviz /nfs/projects/

chown -R tandemviz /nfs/tandemai/

chown -R tandemviz /nfs/home/tandemviz/

15. Update ffdb.sh on Backend node

vi /software/aws/ffdb/ffdb.sh

update "source /software/miniforge3/bin/activate apps_py38" to "source /software/miniforge3/bin/activate ta_force"

add "# load module" before "# init database"

load module

module load tabase/py38

module load taworkflow/v20_py38

```
# cp files
/bin/cp -f /software/aws/ffdb/$projectname/ffengine.cfg Force
/bin/cp -f /software/aws/ffdb/$projectname/alembic.ini Force

# load env params
source /etc/profile
source /software/miniforge3/bin/activate ta_force

# evn params setting
export PYTHONPATH=/software_common/ta_ffengine_latest:$PYTHONPATH
export FFENGINE_PKG_PATH=/software_common/ta_ffengine_latest/ffengine
export FFENGINE_HOME=$projectpath/$projectname/Force

# load module
module load tabase/py38
module load taworkflow/v20_py38

# init database
/software_common/ta_ffengine_latest/ffengine/bin/ffde initdb -a $projectpath/$projectname/Force/alembic.ini

# update the owner and permission
cd "$projectpath/$projectname"
#/bin/chown -R tandemviz:$projectname Force*
#/bin/chmod -R g+w Force_Data
#/bin/chmod -R g+w Force/alembic.ini
sudo /usr/bin/chown -R tandemviz:$projectname Force*
```

16. Update sssd configure for HeadNode

16.1 get original uid of "tandemviz"

getent passwd tandemviz

16.2 update /etc/sssd/sssd.conf configure to your own AD

16.3 make it effective:

systemctl restart sssd

16.4 clean sssd cache:

sss_cache -E

16.5 check if the user "tandemviz" get new uid

getent passwd tandemviz

17. Recreate slurm user on HeadNode

sacctmgr delete user tandemviz -i

sacctmgr create user tandemviz account=pcdefault admin=Administrator -i

18. Start viz services on Backend node

/usr/local/bin/ansible-playbook --connection=local /opt/tandemai/ansible/ami_startvizservices.yml

19. Project Creation:

19.1 Create AD project groups that correspond to Viz projects.

– Add the "tandemviz" account to these groups to grant access.

19.2 Viz operations:

– Create projects in Viz that map to these AD groups.

20. User Creation :

20.1 AD operations:

– Create AD user accounts for all users who need access to Viz.

– Assign users to the appropriate AD project groups.

20.2 Viz operations:

– Create user accounts in Viz that correspond to these AD users.

– Add users to the appropriate Viz projects based on their AD group memberships.

Cluster

Whether to create a cluster in this stack

Whether to create a cluster in this stack. This option is used when upgrading the stack version.

MPS3BucketName

Marketplace S3 Buket Name of CloudFormation Template. It's filled automatically by AWS.

MPS3BucketRegion

Marketplace S3 Buket Region of CloudFormation Template. It's filled automatically by AWS.

MPS3KeyPrefix

Marketplace S3 Buket Region of CloudFormation Template. It's filled automatically by AWS.

Non-reply Email Account

Email address for the non-reply email. This email is used for OTP and system notifications.

FROM email address

The FROM email address for the non-reply email.

Non-reply Email Password

Email password for the non-reply email account.

SMTP host for non-reply email

The host of the account. The default value is ***smtp.office365.com***. Any other SMTP server works.

SMTP port

The port for connecting to the server. The default value is 587.

TandemVizAMI

This is the alias of the Marketplace AMI that will be deployed as part of this stack. It's filled automatically by AWS.

3.3. Protecting critical resources with Stack Policy

Stack policy is a crucial component for safeguarding your AWS CloudFormation stacks. It is specifically designed to prevent unintended updates or deletions to key resources such as RDS, EFS, and Active Directory, ensuring the stability and integrity of your TandemViz™ environment during updates.

By applying this policy:

- **Critical resources are protected**: Updates that could disrupt services are prevented for resources such as your Elastic File System (EFS), Active Directory, and the RDS instance used by VizDatabase.
- **Flexibility in updates**: While the policy restricts certain actions to prevent accidental service interruptions, it can be modified as needed to allow for maintenance or upgrades, then reapplied to reinstate protections. How to Apply the Stack Policy
- Before launching or updating your stack, set the stack policy by uploading the stack_policy.json as shown in the screenshot below.
- For existing stacks, update the stack policy using the AWS CLI with the aws cloudformation set-stack-policy command. This preventive measure is an essential step in maintaining the operational integrity of your TandemVPC deployment, ensuring that updates can be performed safely without risking critical components of your infrastructure.

```
{
   "Statement": [
     {
       "Effect": "Deny",
       "Principal": "*",
       "Action": [
         "Update:Delete",
         "Update:Replace"
       ],
       "Resource": [
         "LogicalResourceId/EFS",
         "LogicalResourceId/ActiveDirectory",
         "LogicalResourceId/VizDatabase",
         "LogicalResourceId/Opensearch",
         "LogicalResourceId/SlurmDatabase",
         "LogicalResourceId/SSHKey"
       ]
     },
     {
       "Effect": "Allow",
       "Action": "Update:*",
       "Principal": "*",
       "Resource": "*"
     }
   ] }
```

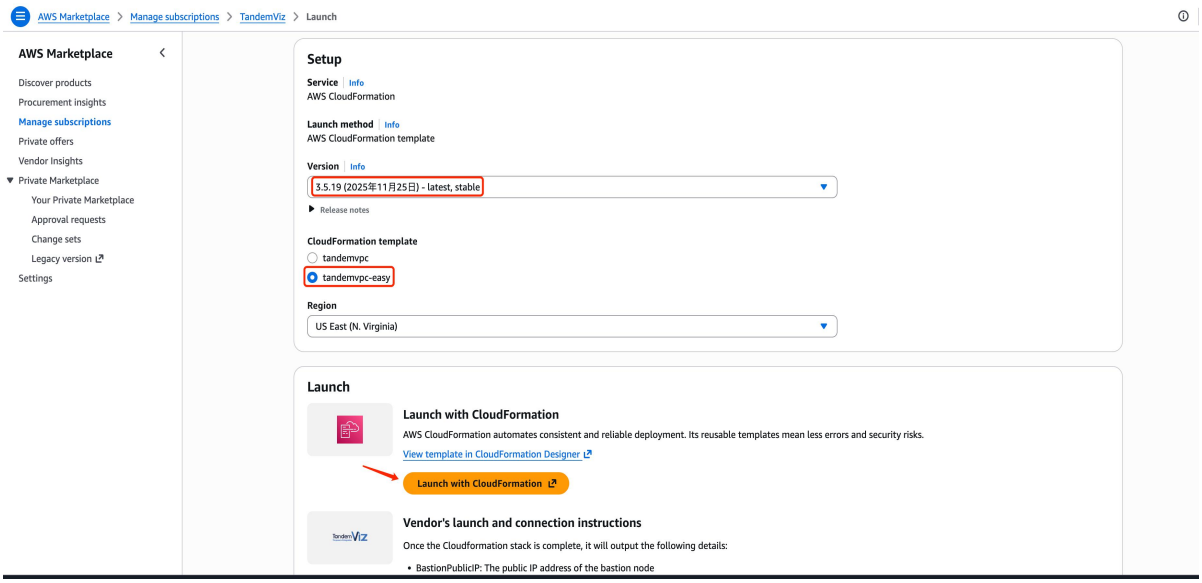
3.4. Version Upgrade

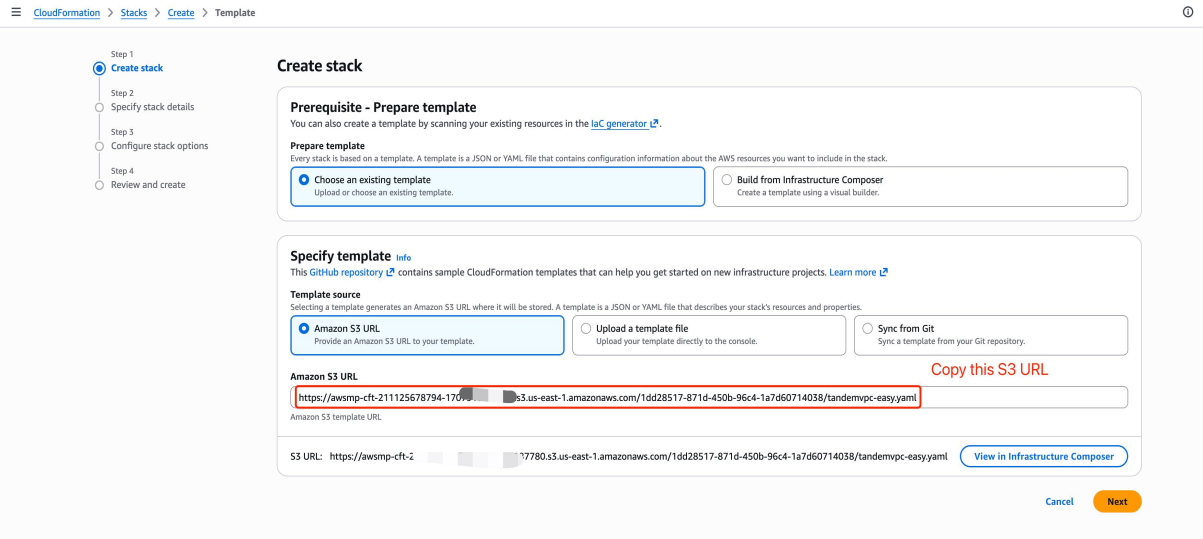
Patches and updates will be available via Cloudformation changesets. For each new version, we will provide a changeset so that you can apply the update for your current stack. These updates include newer versions of TandemViz components and any new infrastructure changes. It is not recommended to manually modify the infrastructure created by TandemVPC.

The upgrade requires four steps:

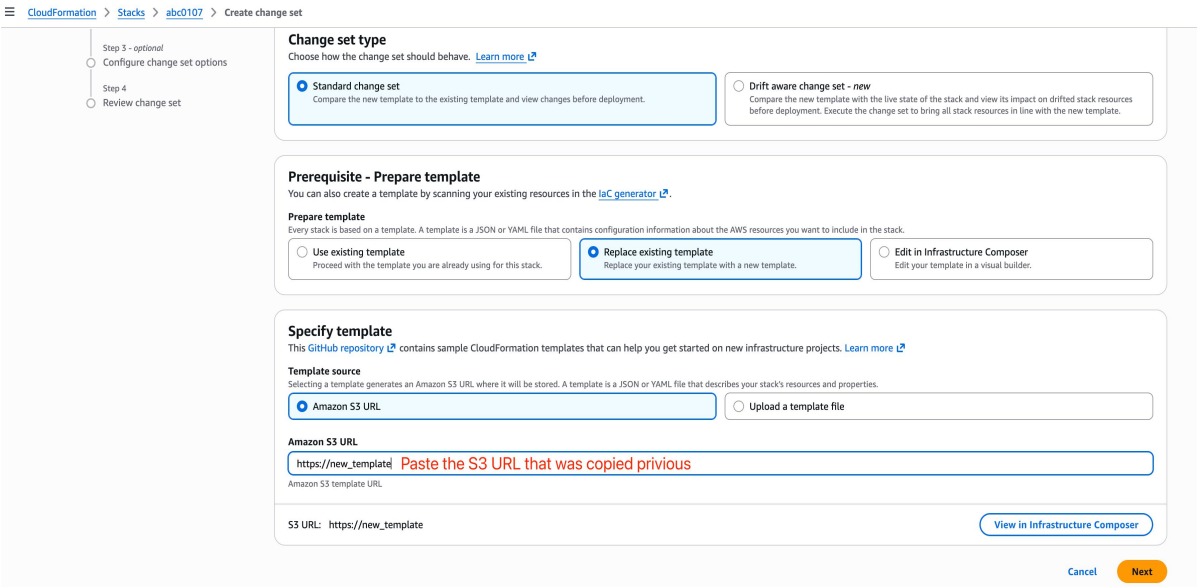
Note: Please keep all parameters with default values.

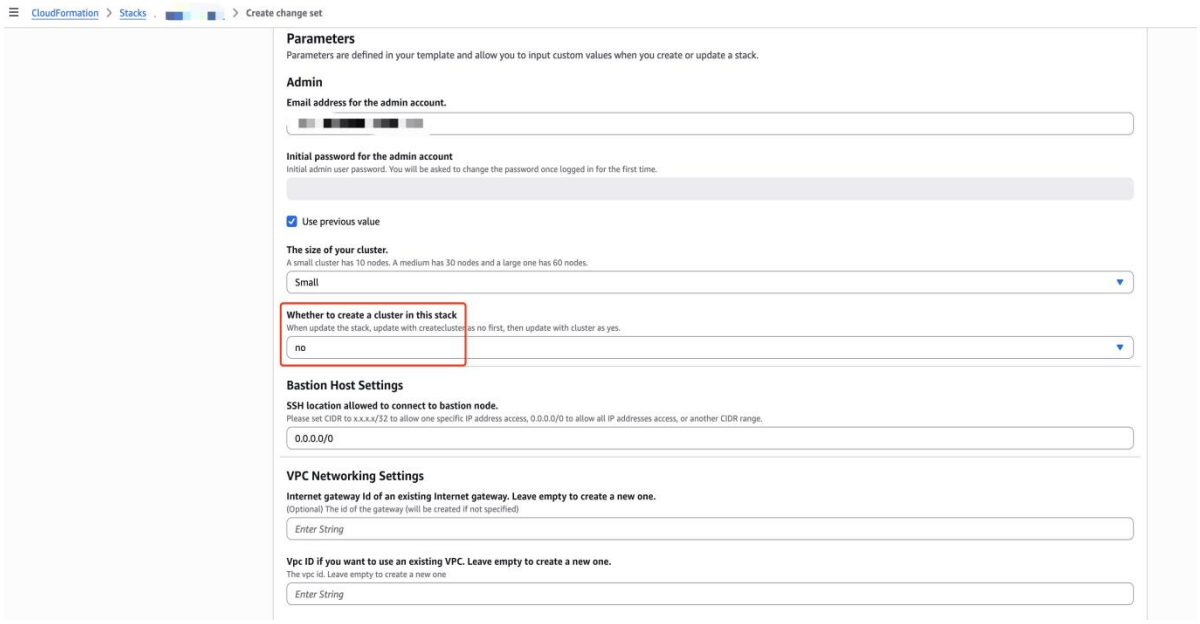
- First, you need to get Amazon S3 URL of new version from Marketplace, then copy it.

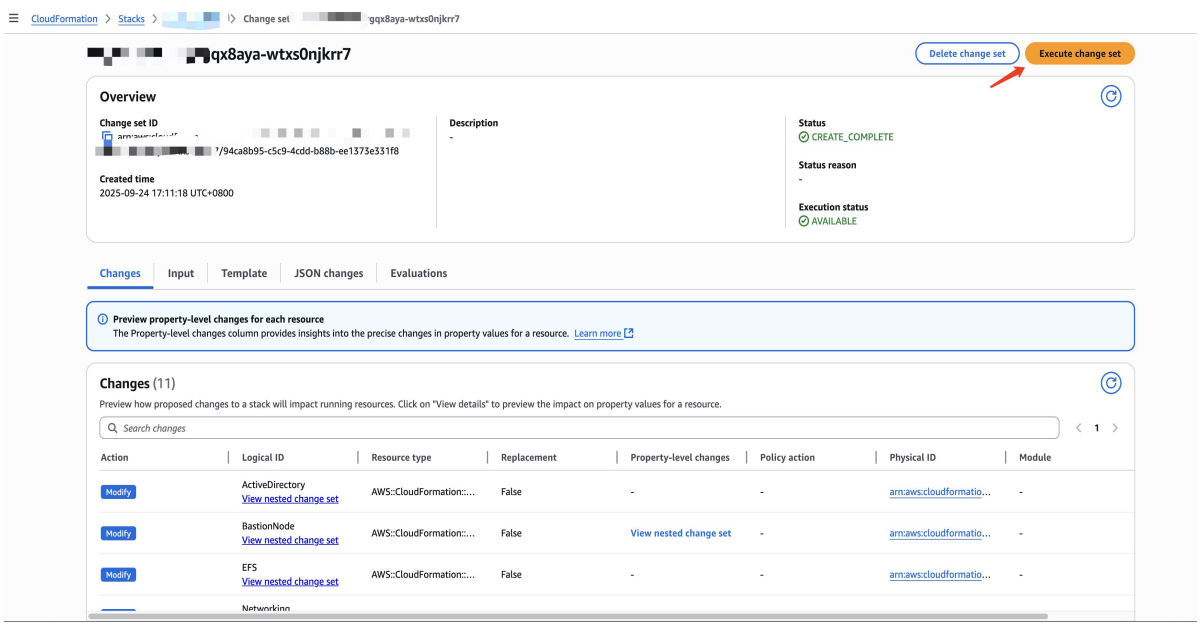


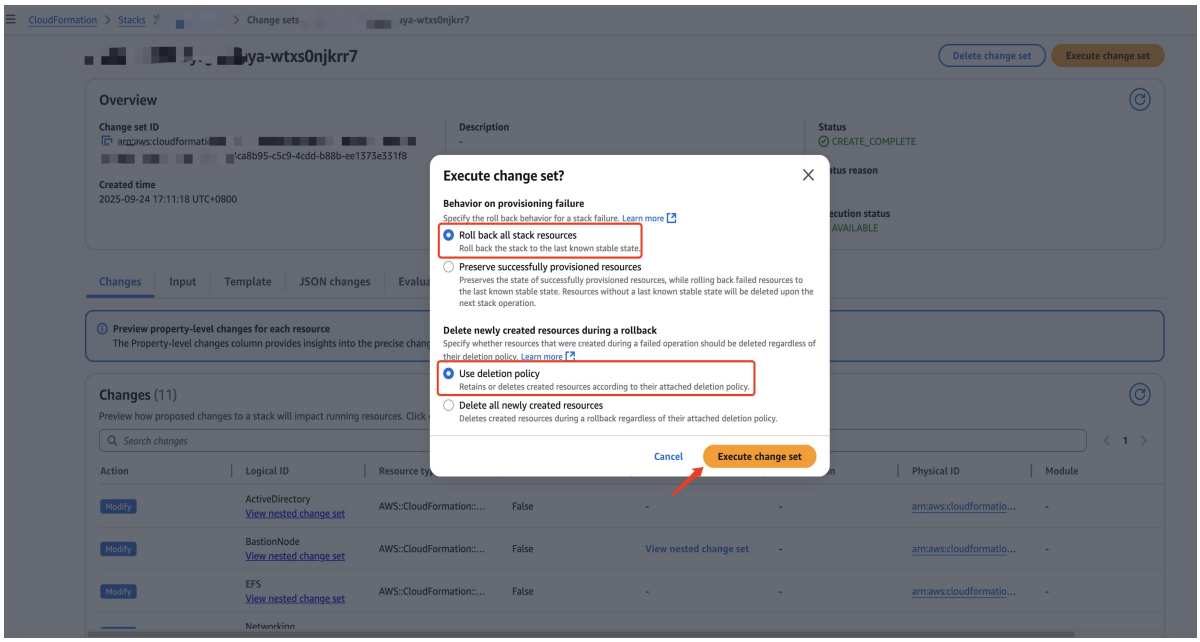


– Second, you need back to original stack then create a "change set" with the new version with the parameter "**Whether to create a cluster in this stack**" to "no". This will essentially destroy the parallel cluster. Your data is intact.

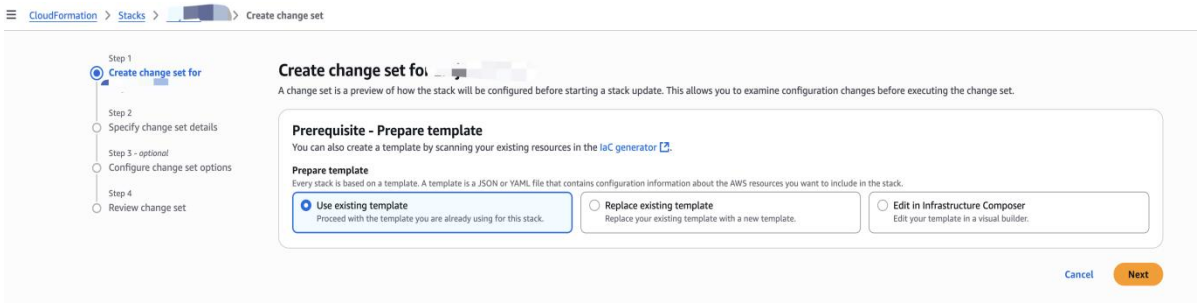


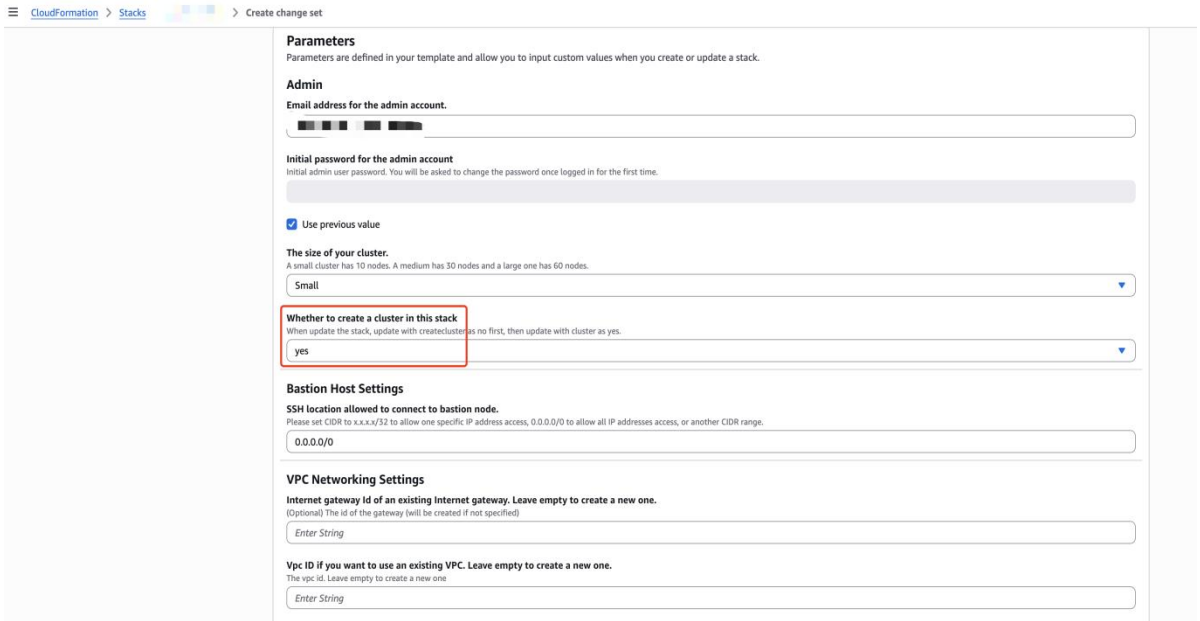

CloudFormation > Stacks > [redacted] > Create change set
Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.
Admin
Email address for the admin account.
[redacted]
Initial password for the admin account
Initial admin user password. You will be asked to change the password once logged in for the first time.
Use previous value
The size of your cluster.
A small cluster has 10 nodes. A medium has 30 nodes and a large one has 60 nodes.
Small
Whether to create a cluster in this stack
When update the stack, update with createcluster as no first, then update with cluster as yes.
no
Bastion Host Settings
SSH location allowed to connect to bastion node.
Please set CIDR to x.x.x.x/32 to allow one specific IP address access, 0.0.0.0/0 to allow all IP addresses access, or another CIDR range.
0.0.0.0/0
VPC Networking Settings
Internet gateway Id of an existing Internet gateway. Leave empty to create a new one.
(Optional) The id of the gateway (will be created if not specified)
Enter String
Vpc ID if you want to use an existing VPC. Leave empty to create a new one.
The vpc id. Leave empty to create a new one
Enter String

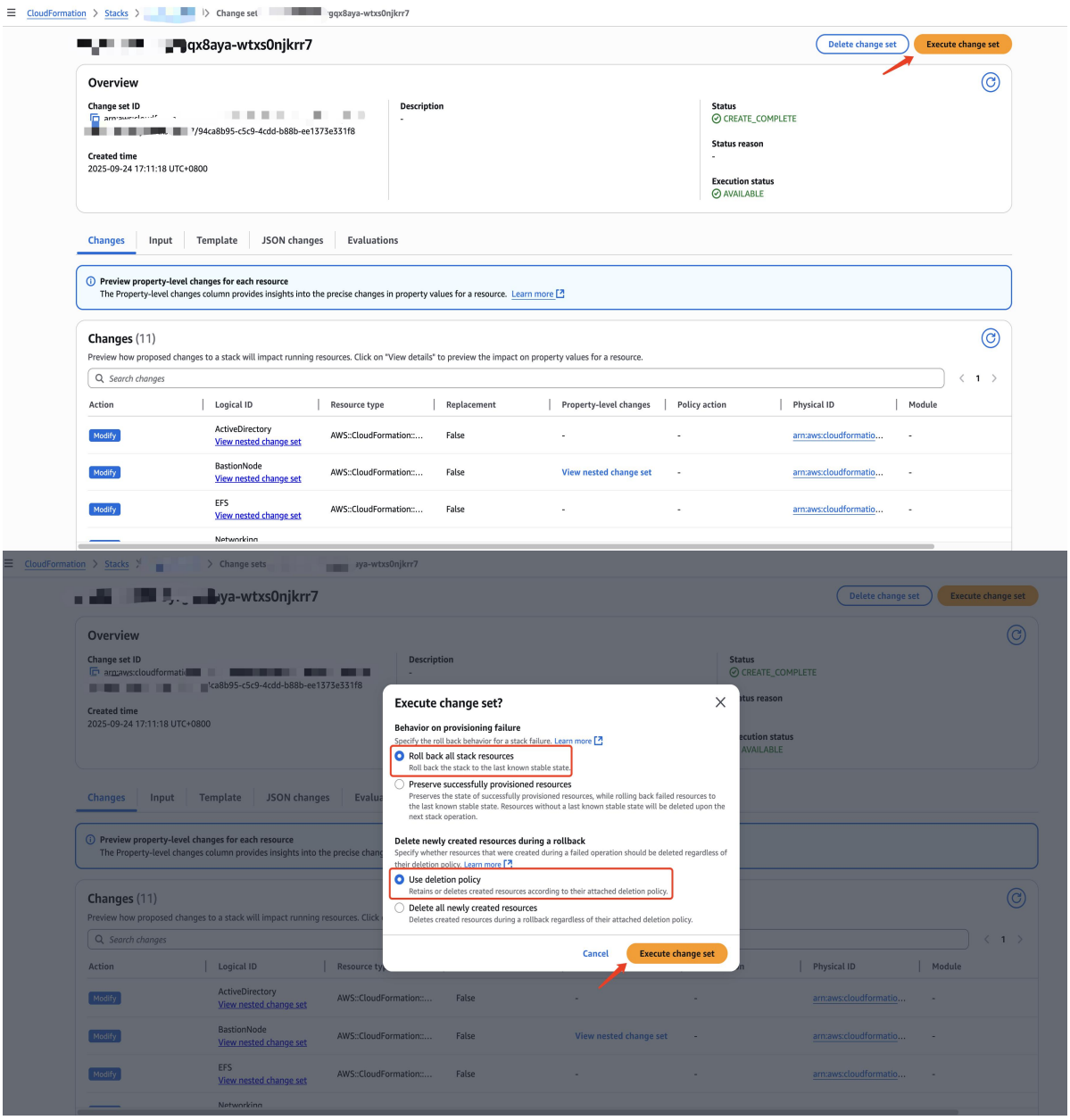

CloudFormation > Stacks > [redacted] > Change set [redacted]gqx8aya-wtxs0njkrr7
[redacted]qx8aya-wtxs0njkrr7
Delete change set
Execute change set
Overview
Change set ID
[redacted]/94ca8b95-c5c9-4cdd-b88b-ee1373e331f8
Created time
2025-09-24 17:11:18 UTC+0800
Description
-
Status
CREATE_COMPLETE
Status reason
-
Execution status
AVAILABLE
Changes
Input
Template
JSON changes
Evaluations
Preview property-level changes for each resource
The Property-level changes column provides insights into the precise changes in property values for a resource. Learn more
Changes (11)
Preview how proposed changes to a stack will impact running resources. Click on "View details" to preview the impact on property values for a resource.
Search changes
1
Action
Logical ID
Resource type
Replacement
Property-level changes
Policy action
Physical ID
Module
Modify
ActiveDirectory
View nested change set
AWS::CloudFormation::...
False
-
-
arn:aws:cloudformatio...
-
Modify
BastionNode
View nested change set
AWS::CloudFormation::...
False
View nested change set
-
arn:aws:cloudformatio...
-
Modify
EFS
View nested change set
AWS::CloudFormation::...
False
-
-
arn:aws:cloudformatio...
-
Networking



– Third, you need to apply the new version with the "**Whether to create a cluster in this stack**" to "yes". This will create a new cluster with all the new packages.







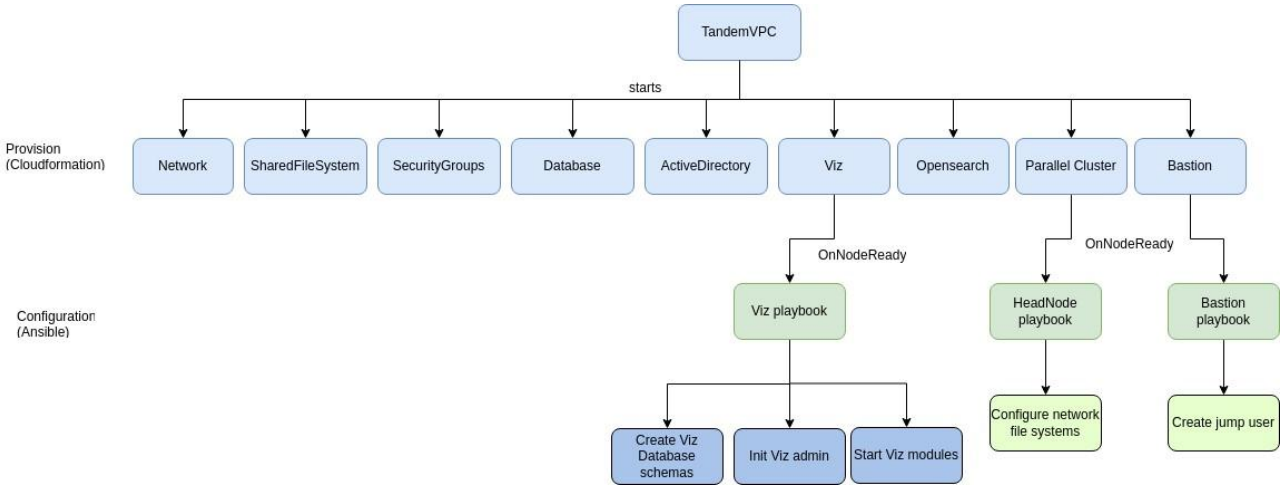
– Forth, we recommand you upgrade the default config of force. Connect to Backend node(Instance type is t3.2xlarg) with “Session Manager”, get DB connection info with command “grep postgres /extravars.yaml”. Then you can connect to DB with command “psql –U t6ndem –h replace_it_with_the_value_of_postgres_host –p 3306 –d tandemviz”, enter

the password with the value of “postgres_root_password”. Then execute the following sql

```
update platform.sys_config set value='[{
  "moduleCode":"force",
  "override":[{"input.qm_software":"full-torsiondrive-psi4",
"slurm_parameter.default.partition":"cpu"}],
  "moduleVersions":[]},
{"moduleCode":"pose",
  "override":[{"slurm_parameter.default.partition":"cpu"}],
  "moduleVersions":[]},
{"moduleCode":"analytics",
  "override":[{"slurm_parameter.default.partition":"cpu"}],
  "moduleVersions":[]},
{ "moduleCode":"atommap",
  "override":[{"slurm_parameter.default.partition":"cpu"}],
  "moduleVersions":[]},
{"moduleCode":"dock",
  "override":[{"slurm_parameter.default.partition":"cpu"}],
  "moduleVersions":[]},
{"moduleCode":"netmap",
  "override":[{"slurm_parameter.default.partition":"lightgpu"}],
  "moduleVersions":[]},
{"moduleCode":"gen",
"override":[{"slurm_parameter.default.partition":"lightgpu"},{"parameter.master.
num_cpus":8}, {"parameter.master.free_cpus":0}, {"parameter.crem.database":
"/nfs/platform/TandemGen/fragments_r2.db"}, {"parameter.core-
hopping.crem_database": "/nfs/platform/TandemGen/fragments_r2.db"}],
  "moduleVersions":[]},
```

```
{"moduleCode":"admet",
  "override":[{"slurm_parameter.default.partition":"cpu"}],
  "moduleVersions":[]},
{"moduleCode":"energy",
  "override":[{"slurm_parameter.default.partition":"project"},
{"parameter.analysis.launch_monitor":false}],
  "moduleVersions":[]},
{"moduleCode":"chemspace",
  "override":[{"slurm_parameter.default.partition":"cpu"}],
  "moduleVersions":[]}]' where id=19;
```

4. Stack Components and Outputs



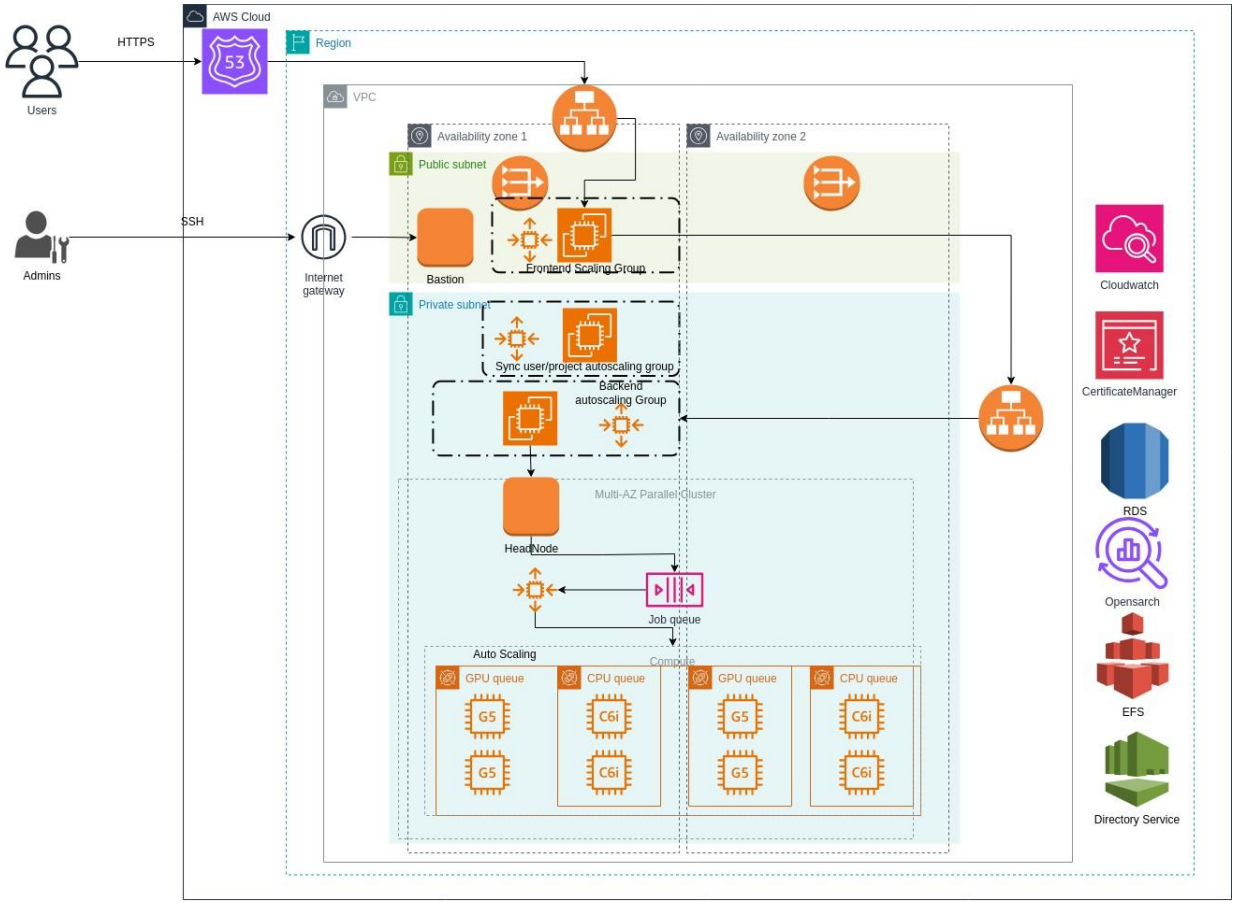
The above diagram shows the various components of TandemVPC. Essentially, it consists of two phases:

- Provision: TandemVPC brings up necessary AWS resources and services such as VPC and its subnets, shared file system, etc. Cloudformation is used to perform this phase.

- Configuration: In this phase, Ansible is triggered by the previous phase to kick off the configuration of TandemViz™ services.

4.1.Provision

The following diagram shows the architecture of TandemVPC once the provision is finished.



At the time of writing this document, TandemVPC can function across multiple AZs in any region. At the center of it is a Parallel Cluster that is optimized for the TandemViz™ workloads. This cluster can send jobs to any selected AZ, while the TandemViz™ frontend and backend only work in the first AZ.

Regions supported

At the time of writing this document, we support all the US and China regions.

Completion time

The stack takes less than one hour to finish provision and configuration.

Configuration

At the last step of the provision phase, TandemVPC kicks off Ansible playbooks to configure various TandemViz™ environments and services.

4.2. Services and components

This part covers various components/services created by TandemVPC that are used by TandemViz™.

Parallel cluster

TandemVPC creates a Parallel Cluster to manage the computation jobs. This cluster uses Slurm as the job scheduler, and uses multiple partitions (as listed below). The cluster will automatically spawn instances (till a specified limit) to serve new jobs and delete unused instances. The Slurm cluster partitions are already described in section 3.1.2.

VPC

If not specified, the deployment will create a new VPC. Otherwise, the deployment will use an existing VPC specified by the customer. Subnets

TandemVPC will create one public subnet and one private subnet for each selected AZ. TandemVPC can support all the AZs available in one region.

Opensearch

TandemViz™ uses OpenSearch to index the molecules generated by Tandem Gen. RDS

TandemVPC uses two serverless RDSs for 1) Slurm Database and 2) TandemViz™ database. While the Slurm database stores the jobs details for the Slurm cluster, the TandemViz database stores details about users, projects and experiment information. User–related data is stored in this TandemViz database. EFS

EFS is used to house all the files generated by Viz's computation. It is mounted into the parallel cluster as a shared file system. All files served from Viz reside in EFS. Directory Service

TandemVPC uses directory service to house user information for the Parallel cluster. SSH key

TandemVPC creates a new SSH key for each deployment. The administrators can use this SSH key to SSH into the EC2 instances.

Once the stack is created, admins are expected to download and store this SSH key from AWS System Manager.

Secrets

TandemVPC creates multiple secrets for various services:

- *ADReadOnlySecret*: this secret is for read only credential for directory service
- *ADADminSecret*: this secret is for admin credential for directory service
- *DomainPrivateKeySecret*: this holds the private key for directory service

- *DomainCertificateSecret*: this holds the certificate secret for directory service
- *OpenSearchSecret*: this holds the secret for Opensearch client
- *DBClusterAdminSecret*: this holds the admin user password for Slurm RDS database
- *DBClusterAdminSecret*: this holds the admin user password for TandemViz™ RDS database
- *VizDBVizSecret*: this holds the user password for TandemViz™ RDS database

CloudWatch

TandemVPC creates 3 CloudWatch log groups for storing and indexing TandemViz™ logs.

- ***/tandemai/{stackname}/tandemviz-frontend:*** this is to store TandemViz frontend Nginx logs.
- ***/tandemai/{stackname}/tandemviz-backend:*** this is to store TandemViz's backend logs.
- ***/tandemai/{stackname}/tandemviz-clustertools:*** this is to store TandemViz's user and group synchronization service. This service syncs the users from TandemViz database to the parallel cluster.

TandemVPC also collects metric data from TandemViz computers (front-end, backend and user-group sync machine). All these metrics and logs are organized into a Cloudwatch Dashboard (named **{stackname}-tandemviz-dashboard**).

TandemVPC also creates multiple alarms to monitor the disk, RAM and CPU usage in TandemViz nodes, as well as the health of the load balancers needed to run TandemViz.

4.3. Testing and troubleshooting

4.3.1. Testing

Once you have deployed the stack without any problem, you can check the stack outputs and visit the ***VizUrl*** to get to the TandemViz home page. From there, you can use your admin password to login. Once projects and users are added (Section 5), you can test submitting jobs from TandemViz.

4.3.2. Troubleshooting

The following table lists solutions to some commonly encountered errors associated with TandemVPC. Most of these errors are caused by unmet requirements.

Problem Description	Solution
Stack failed with error ***The maximum number of addresses has been reached***	This is because you do not have enough elastic IPs in your quota. Consider launching TandemVPC with less available zones (minimal 2) or increase your elastic IP quota.
Stacked failed with error Failed to create AWSServiceRoleForAmazonOpenSearchService	Make sure you check whether you have existing ***AWSServiceRoleForAmazonOpenSearchService*** role exists and select the right option.

4.4 Disaster Recovery

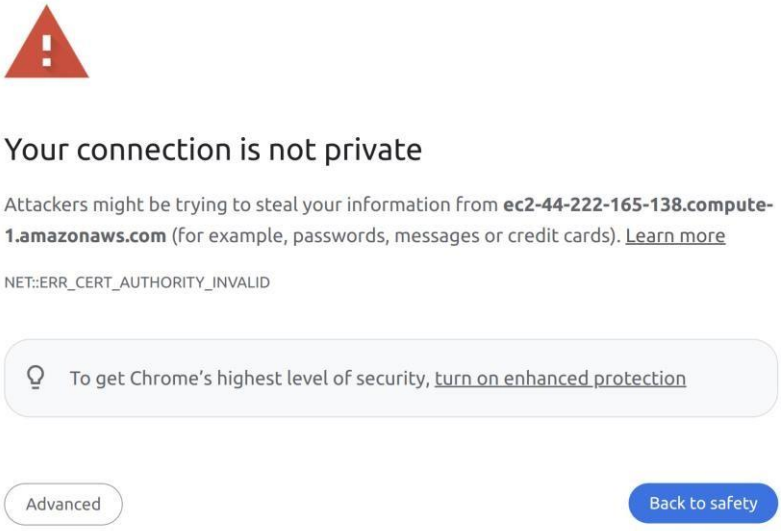
As shown in Section 4.2, TandemVPC uses several managed data stores in AWS such as RDS, Opensearch and EFS for storing TandemViz data. By default, TandemVPC enables automatic snapshots and backup in all those managed services; this means you can always recover these data stores in case of disaster.

5. Product Access Instruction

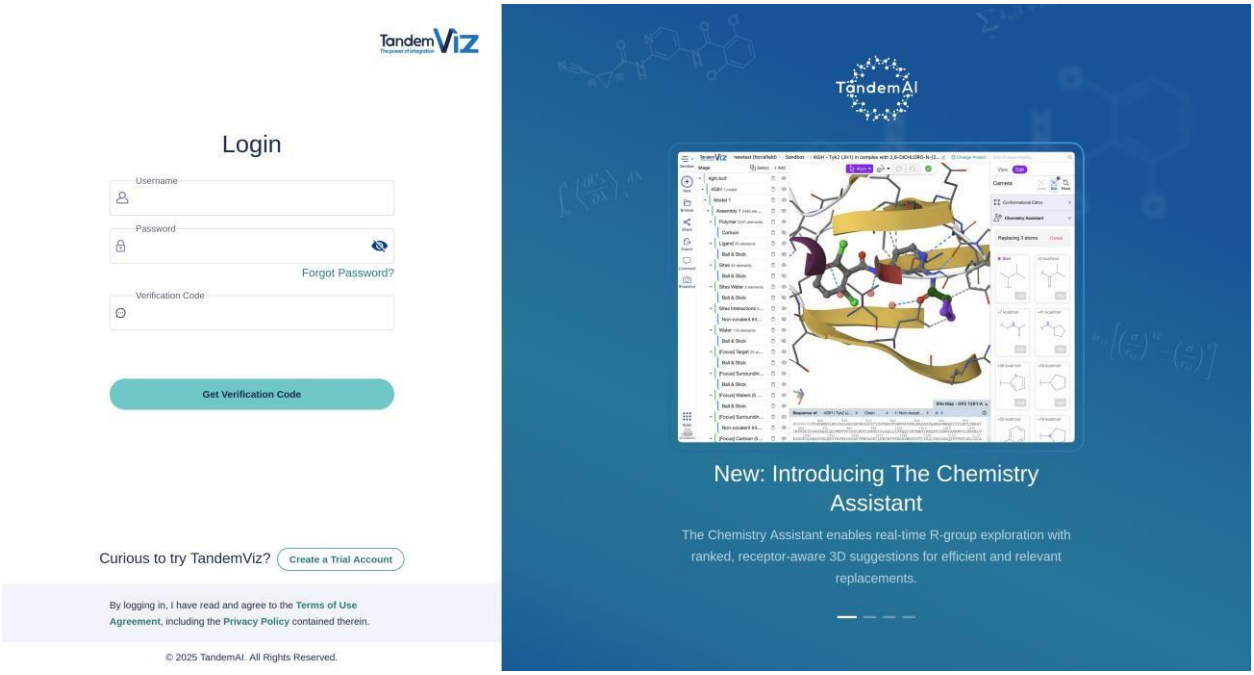
Once the Cloudformation stack is complete, it will output the following details:

- **BastionPublicIP**: The public IP address of the bastion node
- **HeadnodePrivateIp**: The private IP address of the head node of the cluster
- **ClusterCloudwatchDashboardUrl**: The Url to the CloudWatch dashboard of the cluster.
- **TandemVizCloudwatchDashboardUrl**: The Url to the CloudWatch dashboard of TandemViz setup.
- **ParameterStoreKeyId**: The ID of the SSH key stored in System Manager's parameter store
- **VizUrl**: The URL to access TandemViz

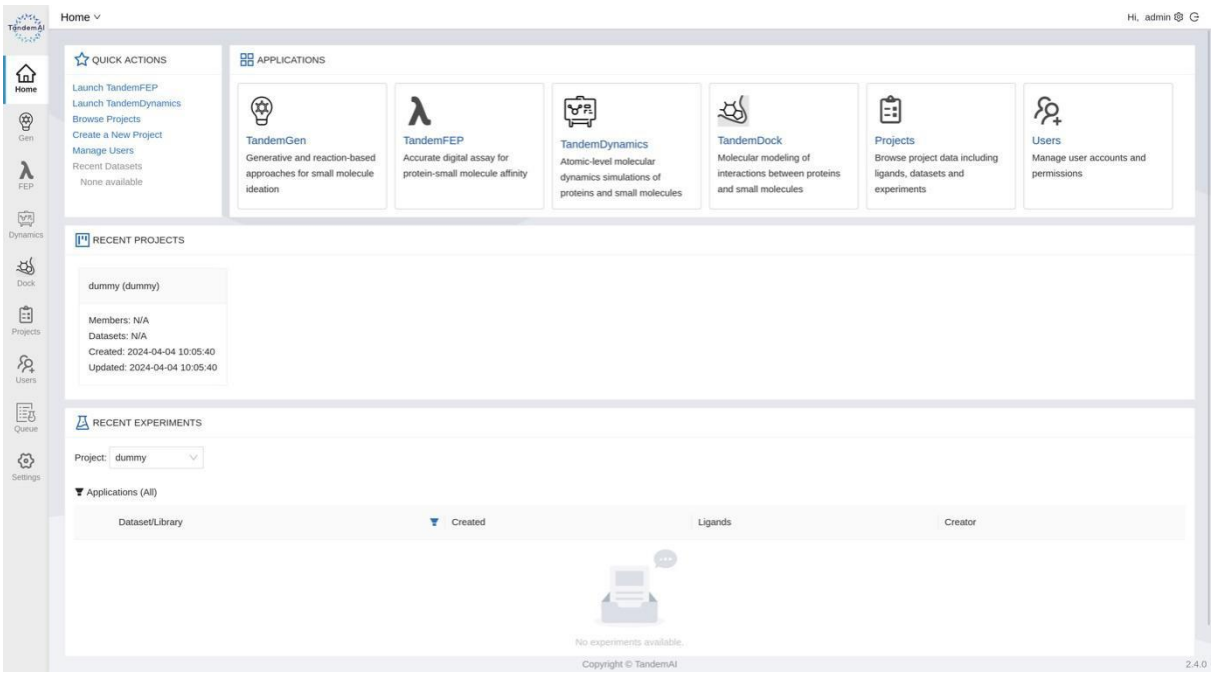
If you do not provide an ARN of the AWS Certificate Manager's public certificate, the first time you visit TandemViz, it will show this sign. This is because a self-signed certificate is used instead of a public certificate.



Once you trust the certificate, it will take you to the TandemViz homepage. Here, you can enter the administrator account using "admin" as username, and the password you entered earlier at the beginning of the deployment. An OTP will be sent to the email you nominated as admin email. From here, you can start adding projects and users.



For the first time login, you will be asked to change the password. This is to make sure your admin password adheres to the platform password standards. From here, you can perform administrative tasks such as adding new projects, adding new users and configuring software packages.



Creating new projects

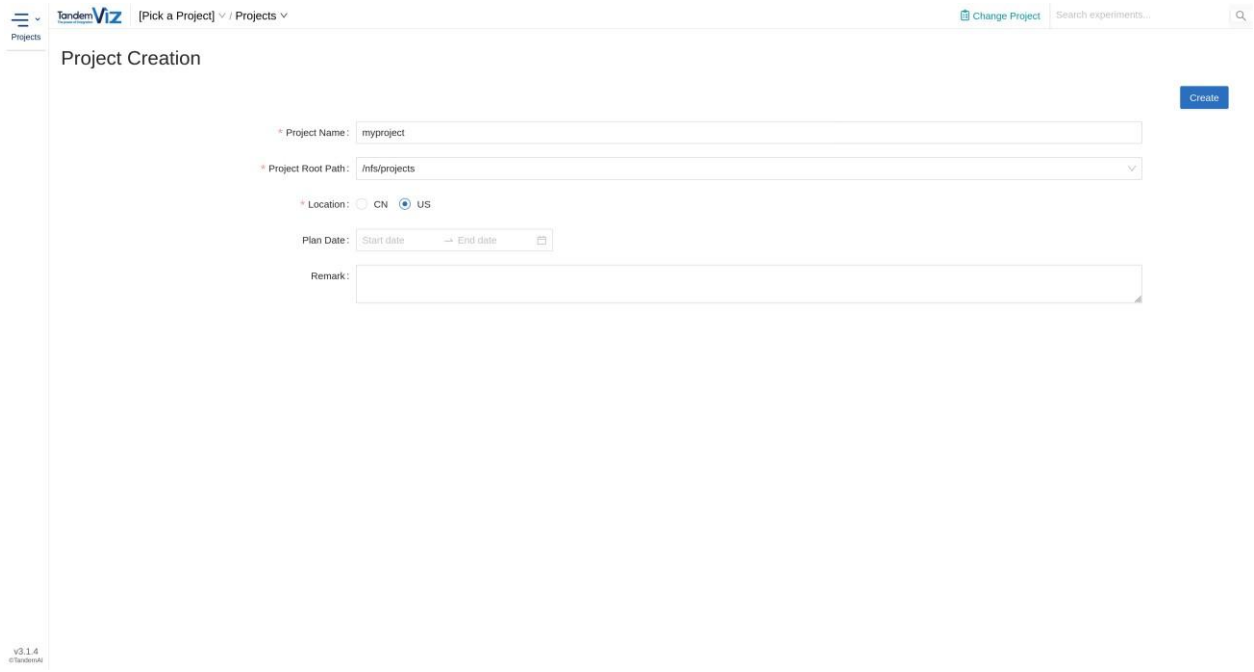
You can click on the ***Project*** button on the left panel to navigate to the project list.

To add a new project, click the ***Create New Project*** button and fill in the project details.

- Project Name: a short name describing project name
- Project Root Path: a root path of all projects. Please keep the default value.
- Location: the location of the project, whether it is US or CN.
- Participant: you can add existing users into the new project. Otherwise, you can add participants later once the project is created.

- Plan Date: the duration of the project. You can leave this field empty.
- Remark: any notes on the project.

There is no restriction on the number of projects you can create.



Adding new users

To add new users, you first need to navigate to the user list by clicking on the ***Users*** button.

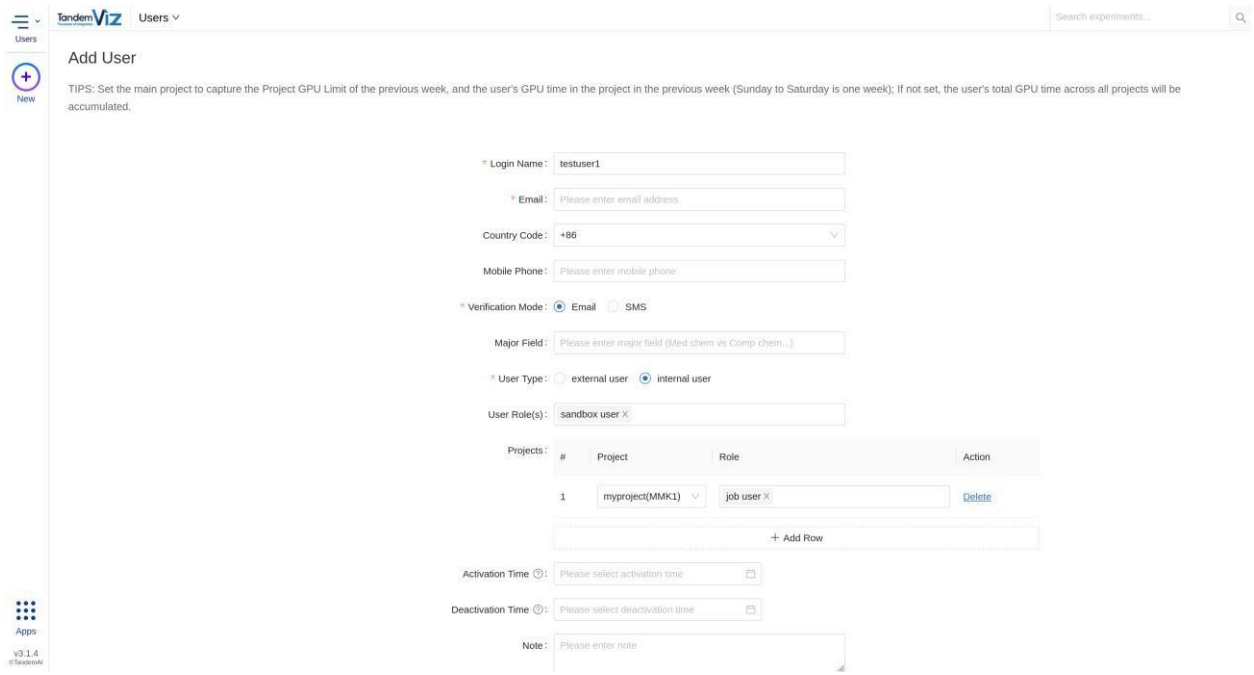
- User Name: the username in the system • Login Name: the name used for logging in
- Email: the user email address.
- Country code: country code of the user's phone number
- Mobile phone: the mobile phone of the user's phone number

Note that we do not use phone numbers for OTP so you can enter any phone number here.

- Verification Mode: Please always select email

- Major Field: any free text
- Project(s): list of projects this user is working on
- Main Project: the user's main project
- User Type: external or internal. Internal users have more control, such as the run parameters.
- User Role(s): the roles of this user. For normal user, please select "job user".
- Activation time: when this account is enabled • Deactivation time: when this account is disabled
- Note: any comment on the user account.

Once the new account is created, make sure you unlock this account. Otherwise, the user cannot log in.



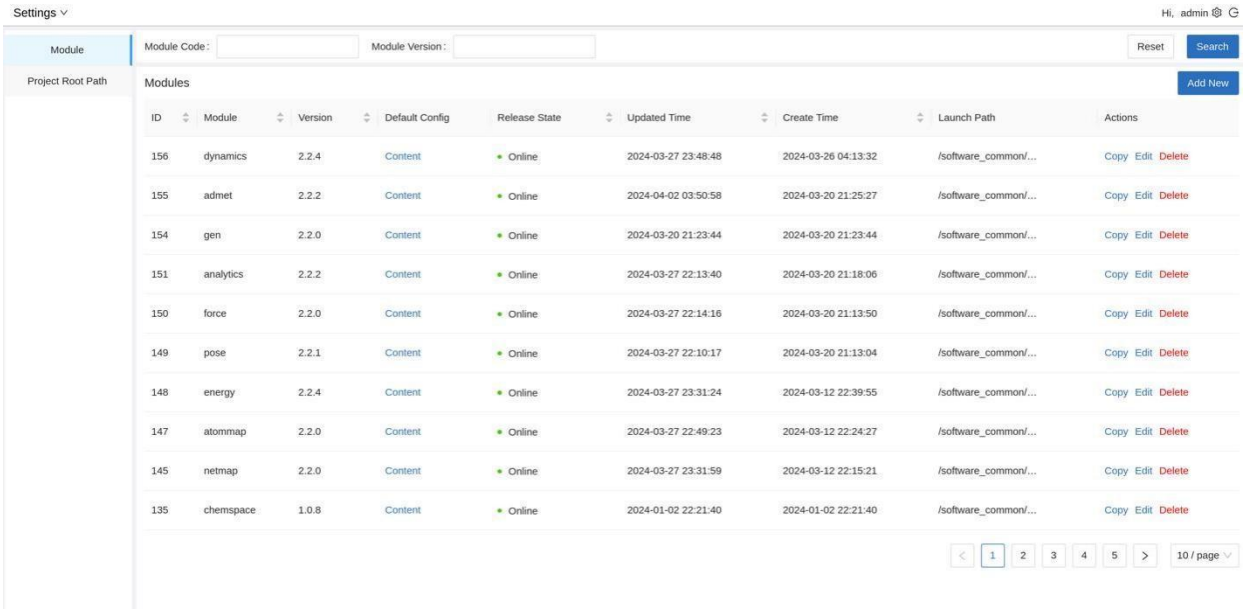
Once the account is created, both the user and administrator will receive emails notifying them about the new account's creation. These emails include details on how to log in, and the initial credentials for the first log in.

Note that there is a limit on the number of users you can create. This is written in the contract. If you reach the user limit, the newly created accounts cannot be used to submit jobs.

System settings

As an admin, you can configure the modules being launched by TandemViz. From the ***Settings*** menu, you can change the default tools configurations, such as default parameters, default Slurm partitions for the jobs.

Module settings



You can modify any default TandemViz modules from this menu. Simply click on the module and you can edit the default parameters.

As an instance, if you want to launch jobs in the project-spot (using spot instances), you can change the ***partition*** variable (path: parameter.slurm_parameter.default.partition) from “project” to “project-spot”.

Project Root Path

This is used to modify the project root path in case you have more than one project mount points. Since TandemVPC only uses a single Elastic File System (EFS) for storing project data, please ***do not*** modify these values.

6. Cost Estimates

This section estimates the cost of 1) infrastructure running TandemViz™ and 2) compute resources of running benchmark cases.

TandemViz Infrastructure Cost

Service Name	Status	Upfront cost	Monthly cost	Description	Region
Amazon EC2	-	0.00 USD	60.30 USD	TandemViz backend	US East (N. Virginia)
Amazon EC2	-	0.00 USD	16.86 USD	Tandemviz frontend	US East (N. Virginia)
Amazon EC2	-	0.00 USD	74.64 USD	Paralel Cluster headnode	US East (N. Virginia)
Amazon EC2	-	0.00 USD	1.53 USD	Bastion node	US East (N. Virginia)
Amazon Aurora MySQL-Compatible	-	0.00 USD	331.84 USD	Parallel cluster database	US East (N. Virginia)
Amazon Aurora PostgreSQL-Compatible DB	-	0.00 USD	413.68 USD	Tandemviz database	US East (N. Virginia)
Amazon OpenSearch Service	-	0.00 USD	223.48 USD	Opensearch for indexi...	US East (N. Virginia)
Elastic Load Balancing	-	0.00 USD	114.61 USD	Load balancers	US East (N. Virginia)
Amazon Virtual Private Cloud (VPC)	-	0.00 USD	80.48 USD	VPC NAT, public IPs	US East (N. Virginia)

The table above shows an estimate of the infrastructure cost for running TandemViz. Note that we ***do not include*** the cost for computation and storage incurred from running computational experiments. This is because this cost varies greatly from client. In the next section, we will show the computing cost related to a benchmark experiment. The cost for running Tandemviz is estimated to be around $1300/month. You can visit here for more details about the estimate.

Computing Cost of a benchmark case study

To provide a guideline for the computing cost in AWS, we have performed a benchmark on our proprietary FEP workflow. This is the most computing intensive and thus the most expensive application in TandemViz. For this benchmark, we perform a single edge FEP calculation on two well-known targets, as shown in the table below.

Protein	System size (Natoms)*	Perturbations (lambdas * direction)	REMD On/Off
TYK2	37K	12	On
BACE	38K	12	On

* Only 1 edge is considered. Solvation included

We then perform the same runs on different types of GPUs, the costs for those runs are shown in the two tables below. All the costs were calculated on 3rd Jan 2024.

Configuration	**Instance**	**Combined throughput <ns/day>**	**Walltime (hours)**	**GPU hour**	**AWS-US**
4 T4 GPUs	g4dn.12xlarge *	29.79	3.652	25.98	$28.02
4 A10G GPUs	g5.12xlarge	84	2.01	11.12	$17.20
4 L4 GPUs	g6.12xlarge	46.35	1.68	14.46	$16.79
4 V100 GPUs	p3.8xlarge*	88.51	1.99	11.5	~$39.99

4 A10G GPUs Spot Instance	g5.12xlarge	83.85	2.16	11.04	$7.83

TYK2 / 37K atoms / 1 edge / 12 lambdas/ 5 ns TI sim/REMD on benchmark

Configuration	Instance	Combined throughput <ns/day>	Walltime (hours)	GPU hour	AWS-US
4 T4 GPUs	g4dn.12xlarge *	25.97	3.86	27.34	$29.50
4 A10 GPUs	g5.12xlarge	79.79	1.86	11.4	$17.50
4 L4 GPUs	g6.12xlarge	46.35	1.7	13.81	$16.04
4 V100 GPUs	p3.8xlarge*	84.32	1.98	11.59	~$40.30
4 A10G GPUs Spot Instance	g5.12xlarge	79.62	2.05	11.36	$8.02

BACE / 38K atoms / 1 edge / 12 lambdas / 5 ns RI sim/REMD on

Licensing Cost

TandemVPC will be available via AWS marketplace with contract-based pricing model. The pricing brackets are listed below:

- $100K per user 1–2 users
- $80K per user 3–10 users
- Please contact us for pricing for more than 10 users.

7. Data Storge, Encryption and Backup/Restore

TandemViz™ stores data using 3 services

- Elastic File Storage (EFS)

This storage is used to store user inputs, and the outputs from the computation started by TandemViz™. EFS is mounted to all the Parallel Cluster nodes, and to the TandemViz™ backend node. This POSIX file system has permissions defined in the Directory Service created by this stack.

- RDS

This stack creates two serverless database clusters: one for storing SLURM database and one for storing TandemViz™ database.

The TandemViz™ database stores all the user, project data and metadata and the jobs details (input parameters, jobs configuration, and outputs).

- Directory Service

This service stores the project and usernames in the system.

- Opensearch

The TandemViz™ uses Opensearch to index all molecules generated by a generative AI tool called TandemGen. This indexing is then used to speed up the exploration (search, filter, sub-setting) of molecules.

All these managed services are backed up regularly and can be restored in case they go down.

8. Support

8.1. Receiving support

All support requests should be emailed to TandemViz-Support@tandemai.com. Our customer service representatives will field the support questions to the appropriate department. Tiers of support are detailed below.

8.2. General and Technical support tiers

Tier 0: (Self-service) Self-service portals such as FAQs and knowledge bases. The customer can independently resolve issues here. Examples include "How do I upload a ligand?" and similar questions that do not require a TandemAI employee.

Tier 1: (TandemViz-centric) Requires a TandemViz™ trained internal support engineer. Questions are more complex than in Tier 0 and are isolated from OS/HPC code/infrastructure. An example could be "I gave read permissions to a project to my scientist, but she cannot view the project page?"

Tier 2: (OS-centric/Cloud infrastructure-centric) Requires an OS and/or HPC trained internal support engineer because software issues in this part are not TandemViz™ specific or scientific by nature. Examples include "Viz logs state that a simulation file does not exist, what went wrong?" or "A TandemViz™ log says that a job failed to submit, what went wrong?" or "Jobs are not submitting because the cluster head node is down"

Tier 3: (AppSci-centric) These are complex scientific questions where Application Scientists can provide answers. Questions can include "My simulation fails in the alchemical simulation?" or "I can't seem to dock a ligand in Pose and need help understanding why".

External Requirement	Criteria
Services provided	TandemViz™ with cloud compute cluster backend running modules for an FEP

	calculation, Generative design, and ADMET properties.
Email communication	12 hour response window
Critical issue support hours (email)	12 hour response window
Major issue support hours (email)	12 hour response window
Scientific issue support hours (email)	24 hour response on weekdays
Non-critical/major issue support hours (email)	24 hour response on weekdays
Core data backup retention	AWS RDS persistent storage backed up daily on AWS
Intermediate file data storage	AWS EFS persistent storage backed up daily on AWS
Data version compatibility	Backwards compatible for previous version only
Software updates	TandemAI will prepare and make available AMIs and update procedures with every new release. Client admins need to update their stack to desired version.
Data security	Adhere to ISO 27001 standards and practices

8.3. Definitions

8.3.1. Bug severity

Low: Product will operate without degraded service.

Minor: Users experience some intermittent degraded behavior

Major: Customer can use the product. Specific runs however are failing computation due to OS system issues and not scientific issues.

Critical: Customer cannot use most or all of the product.

8.3.2. Data

Core: This data is required to populate fields in TandemViz™ and allow the user to interact with the product. Examples include the Viz database.

Intermediate: This data is intermediate run files produced by computation on the compute cluster. Examples include simulation files, program files, and similar intermediate files.

8.4. Service Level Agreement (SLA)

1. *Introduction*

This Service Level Agreement ("SLA") is entered into between TandemAI ("Provider"), and [Customer Name] ("Customer"), effective as of [Effective Date]. This SLA outlines the terms and conditions for the provision of services by the Provider to the Customer.

2. *Services Provided*

The Provider agrees to provide the following services to the Customer:

- TandemViz web-application deployed into Customer Amazon Web Services (AWS) Virtual Private Cloud (VPC).
- TandemOS compute system deployed with TandemViz into Customer AWS VPC with AWS Parallel Cluster (PCluster) compute system capable of running modules for an FEP calculation, generative design, and ADMET properties.
- Infrastructure software to provision hosts and services in Customer AWS VPC and software updates.

3. *Service Availability*

The Provider agrees to maintain a minimum level of service availability of 95% response to reported issues on infrastructure hosted and TandemViz access during the agreed service hours, excluding scheduled maintenance periods.

4. *Response Time*

The Provider agrees to respond to any service-related inquiries or incidents within 12 hours of notification during the agreed service hours for 95% of all inquiries.

5. *Performance Metrics*

The Provider agrees to regularly measure and report on the following performance metrics:

Metric Description	Metric value
Critical Issue Resolution	Resolution within 72 hours
Major Issue Resolution	Resolution within 96 hours
Scientific issue inquiry	Correspondence within 12 hours
Data version compatibility	Backwards compatible for previous version only
Core data backup retention	Daily via AWS
Intermediate file storage	Subject to Customer AWS retention policy
Software updates	TandemAI will prepare and make available AMIs and update procedures with every new release. Client admins need to update their stack to desired version.

6. *Maintenance and Downtime*

The Provider reserves the right to schedule maintenance activities that may result in temporary service unavailability. The Provider agrees to notify the Customer in advance of any scheduled maintenance. 7. *Governing Law*

This SLA shall be governed by and construed in accordance with the internal laws of the State of New York, without giving effect to any choice of law or conflict of law provision or rule.

8. *Entire Agreement*

This SLA constitutes the entire agreement between the parties with respect to the subject matter hereof and supersedes all prior and contemporaneous agreements and understandings, whether written or oral, relating to such subject matter.

9. *Amendments*

This SLA may be amended or modified only in writing and signed by both parties.

Appendix A — Base AIM Policy for TandemVPC

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Action": [
          "ec2:Describe*"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "EC2Read"
      },
      {
        "Action": [
          "ec2:AllocateAddress",
          "ec2:AssociateAddress",
          "ec2:AttachNetworkInterface",
          "ec2:AuthorizeSecurityGroupEgress",
          "ec2:AuthorizeSecurityGroupIngress",
          "ec2:CreateFleet",
          "ec2:CreateLaunchTemplate",
          "ec2:CreateLaunchTemplateVersion",
          "ec2:CreateNetworkInterface",
          "ec2:CreatePlacementGroup",
          "ec2:CreateSecurityGroup",
          "ec2:CreateSnapshot",
          "ec2:CreateTags",
          "ec2:CreateVolume",
```

```
        "ec2:DeleteLaunchTemplate",
        "ec2:DeleteNetworkInterface",
        "ec2:DeletePlacementGroup",
        "ec2:DeleteSecurityGroup",
        "ec2:DeleteVolume",
        "ec2:DisassociateAddress",
        "ec2:ModifyLaunchTemplate",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:ModifyVolume",
        "ec2:ModifyVolumeAttribute",
        "ec2:ReleaseAddress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "ec2:CreateVpc",
```

```
        "ec2:DeleteVpc",
        "ec2:CreateSubnet",
        "ec2:DeleteSubnet",
        "ec2:CreateKeyPair",
        "ec2:DeleteKeyPair"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EC2Write"
    },
    {
      "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:ListTagsOfResource",
        "dynamodb:CreateTable",
        "dynamodb:DeleteTable",
        "dynamodb:GetItem",          "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:Query",
        "dynamodb:TagResource"
      ],
```

```
      "Resource": "arn:aws:dynamodb:*:<AWS ACCOUNT
ID>:table/parallelcluster-*",
      "Effect": "Allow",
      "Sid": "DynamoDB"
    },
    {
      "Action": [
        "route53:ChangeResourceRecordSets",
        "route53:ChangeTagsForResource",
        "route53:CreateHostedZone",
        "route53:DeleteHostedZone",
        "route53:GetChange",
        "route53:GetHostedZone",
        "route53:ListResourceRecordSets",
        "route53:ListQueryLoggingConfigs"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "Route53HostedZones"
    },
    {
```

```
    "Action": [
      "cloudformation:*"
    ],
  "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudFormation"
  },
  {
    "Action": [
      "cloudwatch:PutDashboard",
      "cloudwatch:ListDashboards",
      "cloudwatch:DeleteDashboards",
      "cloudwatch:GetDashboard",
      "cloudwatch:PutMetricAlarm",
      "cloudwatch:DeleteAlarms",
      "cloudwatch:DescribeAlarms",
      "cloudwatch:PutCompositeAlarm"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatch"
  },
  {
    "Action": [
```

```
      "iam:GetRole",
      "iam:GetRolePolicy",
      "iam:GetPolicy",
      "iam:SimulatePrincipalPolicy",
      "iam:GetInstanceProfile"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/*",
      "arn:aws:iam::<AWS ACCOUNT ID>:policy/*",
      "arn:aws:iam::aws:policy/*",
      "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*"
    ],
    "Effect": "Allow",
    "Sid": "IamRead"
  },
  {
    "Action": [
      "iam:CreateInstanceProfile",
      "iam:DeleteInstanceProfile",
      "iam:AddRoleToInstanceProfile",
      "iam:RemoveRoleFromInstanceProfile"
```

```
        ],
        "Resource": [
          "arn:aws:iam::<AWS ACCOUNT ID>:instance-
profile/parallelcluster/*" ],
      "Effect": "Allow",
        "Sid": "IamInstanceProfile"
      },
      {
        "Condition": {
          "StringEqualsIfExists": {
            "iam:PassedToService": [
              "lambda.amazonaws.com",
              "ec2.amazonaws.com",
              "spotfleet.amazonaws.com"
            ]
          }
        },
        "Action": [
          "iam:PassRole"
        ],
        "Resource": [
          "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
        ],
        "Effect": "Allow",
```

```
    "Sid": "IamPassRole"
  },
  {
    "Action": [
      "lambda:CreateFunction",
      "lambda:DeleteFunction",
      "lambda:GetFunctionConfiguration",
      "lambda:GetFunction",
      "lambda:InvokeFunction",
      "lambda:AddPermission",
      "lambda:RemovePermission",
      "lambda:UpdateFunctionConfiguration",
      "lambda:TagResource",
      "lambda:ListTags",
      "lambda:UntagResource"
    ],
    "Resource": [
      "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:parallelcluster-*",
      "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:pcluster-*"
    ],
    "Effect": "Allow",
```

```
    "Sid": "Lambda"
  },
  {
```

```
    "Action": [
      "s3:*"
    ],
    "Resource": [
      "arn:aws:s3:::parallelcluster-*",
      "arn:aws:s3:::aws-parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "S3ResourcesBucket"
  },
  {
    "Action": [
      "s3:Get*",
      "s3:List*"
    ],
    "Resource": "arn:aws:s3:::*-aws-parallelcluster*",
    "Effect": "Allow",
    "Sid": "S3ParallelClusterReadOnly"
  },
  {
    "Action": [
```

```
      "elasticfilesystem:*"
    ],
    "Resource": [
      "arn:aws:elasticfilesystem:*:<AWS ACCOUNT ID>:*"
    ],
    "Effect": "Allow",
    "Sid": "EFS"
  },
  {
    "Action": [
      "logs:DeleteLogGroup",
      "logs:PutRetentionPolicy",
      "logs:DescribeLogGroups",
      "logs:CreateLogGroup",
      "logs:TagResource",
      "logs:UntagResource",
      "logs:FilterLogEvents",
      "logs:GetLogEvents",
      "logs:CreateExportTask",
      "logs:DescribeLogStreams",
      "logs:DescribeExportTasks",
      "logs:DescribeMetricFilters",
      "logs:PutMetricFilter",
```

```
        "logs:DeleteMetricFilter"
      ],
      "Resource": "*",
        "Effect": "Allow",
        "Sid": "CloudWatchLogs"
      },
    {
      "Action": [
        "resource-groups:ListGroupResources"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "ResourceGroupRead"
    },
    {
      "Sid": "AllowDescribingFileCache",
      "Effect": "Allow",
      "Action": [
        "fsx:DescribeFileCaches"
      ],
      "Resource": "*"
```

```
    },
    {
      "Action": "secretsmanager:DescribeSecret",
      "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT
ID>:secret:<SECRET NAME>",
      "Effect": "Allow"
    },
    {
      "Effect": "Allow",
      "Action": [
        "*"
      ],
      "Resource": [
        "arn:aws:es:*:<AWS ACCOUNT ID>:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": [
```

```
            "opensearchservice.<AWS::Region>.amazonaws.com"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:*"
      ],
      "Resource": [
        "arn:aws:rds:*:<AWS ACCOUNT ID>:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:DeleteAutoScalingGroup",
        "elasticloadbalancing:CreateLoadBalancer",
        "elasticloadbalancing:DeleteLoadBalancer",
```

```
      "elasticloadbalancing:CreateListener",
      "elasticloadbalancing:DeleteListener",
      "elasticloadbalancing:CreateTargetGroup",
      "elasticloadbalancing:DeleteTargetGroup"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "kms:CreateKey",
      "kms:DeleteKey",
      "kms:Encrypt",          "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:CreateGrant",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "acm:GetCertificate",
```

```
    "Resource": "arn:aws:acm:<AWS::Region>:<AWS ACCOUNT ID>:certificate/*"
  },
    {
```

```
        "Effect": "Allow",
        "Action": [
            "license-manager:CheckoutLicense",
            "license-manager:ListReceivedGrants",
            "license-manager:GetLicense",
            "license-manager:GetGrant"
        ],
        "Resource": "*"
    }
  ]
}
```

Appendix B – IAM user, role, groups and policy resources created by TandemVPC

Resource Name	Type	Purposes
activedirectory.yaml		
JoinRole	AWS::IAM::Role	This role is created to allow the Active Directory Admin node to join the realm.
flowlog.yaml		
FlowLogRow	AWS::IAM::R	This role is created to allow FlowLog to write logs to CloudWatch.

	ole	
viz.yaml		
VizFrontendRole	AWS::IAM:: R ole	This role is created so that the TandemViz frontend node can publish its logs and metrics to CloudWatch
VizBackendRole	AWS::IAM:: R ole	This role is created so that the TandemViz backend node can publish its logs and metrics to CloudWatch
VizUserGroupManage m entRole	AWS::IAM:: R ole	This role is created so that the TandemViz user group node can publish its logs and metrics to CloudWatch
pcluster_cluster.yaml		
PclusterLambdaRole	AWS::IAM:: R ole	This role is need for Pcluster setup (https://docs.aws.amazon.com/parallelcluster/latest/ug/pclus te r-v3.html)
EventsPolicy	AWS::IAM:: ManagedPo l icy	A managed policy for Pcluster (https://docs.aws.amazon.com/parallelcluster/latest/ug/pclus te r-v3.html) to create, delete, event rules and event targets
S3Policy	AWS::IAM:: ManagedPo l icy	A managed policy for Pcluster to list buckets, its versions and perform all actions on a bucket create by Pcluster
pcluster_policies.yaml		
ParallelClusterLambda Role	AWS::IAM:: R ole	This role is needed for Pcluster setup
ParallelClusterFSxS3A c cessPolicy	AWS::IAM:: P olicy	This policy is needed for Pcluster to access FSx. This policy is **not created** since TandemVPC is not using FSx**.**

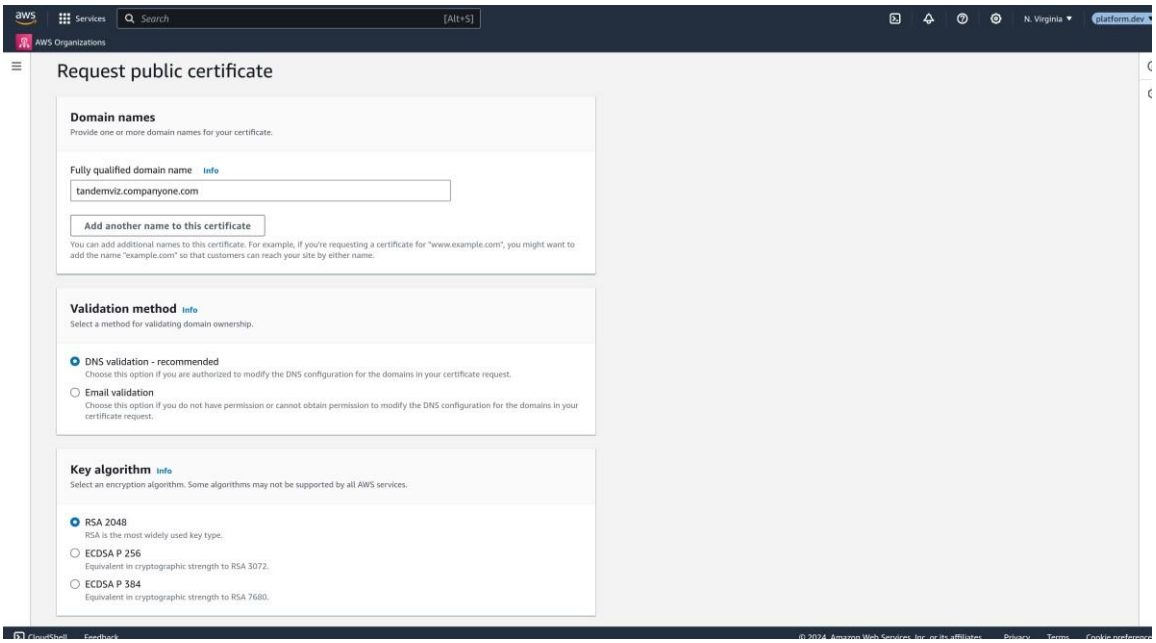
DefaultParallelClusterIa mAdminPolicy	AWS::IAM::ManagedPol icy	A default policy for Pcluster
ParallelClusterClusterP olicyBatch	AWS::IAM::ManagedPol icy	A managed policy for Pcluster to access AWS Batch. This policy is **not created** since TandemVPC is not using Batch**.**
ParallelClusterClusterP olicy	AWS::IAM::ManagedPol icy	A managed policy needed for Pcluster to function
ParallelClusterListIma g esManagedPolicy,	AWS::IAM::ManagedPol icy	A managed policy to allow Pcluster to list images
ParallelClusterDescrib e ImageManagedPolicy	AWS::IAM::ManagedPol icy	A managed policy to allow Pcluster to describe images
ParallelClusterLogRetr i evalPolicy	AWS::IAM::ManagedPol icy	A managed policy for retrieving cluster logs

Appendix C – Using your own DNS

This appendix lists the steps to setup TandemVPC with your own DNS. In this example, the DNS provider is from a third party. If you are using AWS Route 53, it will be very similar, and even simpler.

1. Create a Certificate in AWS Certificate Manager

You need to request a Public Certificate. As an example, we are going to create tandemviz.companyone.com. For validation, we are using DNS validation, which requires you to be able to modify DNS records. The other option is to use email for validation.



2. Verify the certificate request

Once the request is created, AWS Certificate Manager will present you with CNAME name and CNAME value. As an example, these values are:

CNAME name	_5xxxxxxxxxx26e.tandemviz.companyone.com.
CNAME value	_4e6xxxxxxxxxxea.xxxxxx.acm–validations.aws.

You then need to go to your DNS provider, and create a DNS record with the values:

Record Type	CNAME
Host	_5xxxxxxxxxx26e.tandemviz

Value	_4e6xxxxxxxxxxxea.xxxxxx.acm-validations.aws.

Once this DNS record is created, your certificate request should be verified shortly

and the certificate should be created.

3. Create a TandemVPC stack with the Certificate ARN. From the Outputs of the root stack, copy VizURL value.
4. Create another DNS record from your domain name provider

Record Type	ALIAS
Host	tandemviz
Value	VizURL value

Appendix D – Opensource packages

Name	License Type
slurm 18.08.2	GNU General Public License (GPL) V2
G6	MIT License
React	MIT License
ahooks	MIT License
Ant Design	MIT License
Axios	MIT License
Day.js	MIT License
Lodash	MIT License
Moment.js	MIT License
NGL Viewer	MIT License

NProgress.js	MIT License
Plotly	MIT License
Spring Cloud Alibaba	Apache License 2.0
nacos	Apache License 2.0
postgresql 12	PostgreSQL License, a liberal Open Source license, similar to the BSD or MIT licenses.
rdock	GNU Lesser General Public License v3.0
psi4	GNU Lesser General Public License (version 3)
RDKit	BSD 3-Clause "New" or "Revised" License
ProDy	MIT License
TorchANI	MIT License
ubuntu 18.04	Creative Commons Attribution-ShareAlike 3.0 License
prometheus	Apache License 2.0
ganglia	BSD-licensed
ansible	GNU General Public License v3.0
mysql 5.7	MySQL License
Protein-Ligand Interaction Profiler (PLIP)	GNU General Public License v2.0
Split.js	MIT License
CReM	BSD-3-Clause license
Lib-INVENT	Apache-2.0 license
kubernetes	Apache-2.0 license

Meeko	LGPL-2.1 license
AutoDock-Vina	Apache-2.0 license
psi4/resp	BSD 3-Clause License
docker	Apache License 2.0
dock38	apply for license. DOCK 3 is distributed as source code only
OpenSearch	Apache-2.0 license

FastApi	MIT License
Keycloak	Apache-2.0 license
Nacos python sdk	Apache-2.0 license
uvicorn	BSD-3-Clause license
openbabel	GPL-2.0 license
redis	BSD 3-Clause License
mybatis	Apache-2.0 license
jdk8	GNU General Public License, version 2,with the Classpath Exception
redisson	Apache-2.0 license
jasypt	Apache-2.0 license
tanstack/react-query	MIT
buffer	MIT
js-cookie	MIT
papaparse	MIT
react-dnd	MIT
react-error-boundary	MIT
react-hooks-global-state	MIT
react-json-view	MIT
react-plotly.js	MIT
react-resizable	MIT
react-router-dom	MIT
react-window	MIT
regression	MIT
sprintf-js	MIT
Ketcher	Apache License, Version 2.0
absl-py	Apache 2.0
aiobotocore	Apache 2
aiohttp	Apache 2

aiohttp-cors	Apache License Version 2.0
aiosignal	Apache 2.0
alabaster	BSD 3-Clause License
alembic	MIT
annotated-types	MIT License
antlr4-python3-runtime	BSD
anyio	MIT
argon2-cffi	MIT License
argon2-cffi-bindings	MIT
arrow	Apache License 2.0
ase	LGPLv2.1+
astor	BSD-3-Clause
astroid	LGPL-2.1-or-later
asttokens	Apache 2.0
astunparse	BSD
async-lru	MIT License
asynctest	Apache 2
async-timeout	Apache 2
attrs	MIT
autocommand	LGPLv3
Babel	BSD-3-Clause
backcall	BSD 3-Clause License
backports.functools-lrucache	MIT License
bcrypt	Apache-2.0
beautifulsoup4	MIT License
bel-resources	MIT
biopython	Biopython License (BSD 3-Clause License)

bitarray	PSF
black	MIT
bleach	Apache Software License
blessed	MIT
blinker	MIT License
bokeh	Copyright (c) 2012 - 2023 Anaconda Inc. and Bokeh Contributors
boto3	Apache License 2.0

botocore	Apache License 2.0
Bottleneck	Simplified BSD
Brotli	MIT
brotlipy	MIT
cached-property	BSD
cachetools	MIT
captum	BSD-3
certifi	MPL-2.0
cffi	MIT
cftime	License :: OSI Approved :: MIT License
chardet	LGPL
charset-normalizer	MIT
cheroot	BSD 3-Clause License
CherryPy	BSD 3-Clause License
click	BSD-3-Clause
click-plugins	New BSD
cloudpickle	BSD-3-Clause
cmaes	MIT License
colorama	Copyright 2010 Jonathan Hartley
colorful	MIT License

colorlog	MIT License
colour	BSD 3-Clause License
comm	BSD 3-Clause License
ConfigArgParse	MIT
contextlib2	PSF License
contourpy	BSD 3-Clause License
copier	MIT
coverage	Apache 2.0
cppcheck	GPL
cpplint	BSD-3-Clause
cramjam	MIT
crem	Apache License 2.0
cryptography	Apache-2.0 OR BSD-3-Clause
cvxopt	GNU GPL version 3
cycler	BSD

Cython	Apache
cytoolz	BSD
dacite	MIT
dask	BSD
dataclasses	Apache
datamol	Apache
debugpy	MIT
decorator	new BSD License
deepchem	MIT
deepspeed	MIT
defusedxml	PSFL
dill	BSD-3-Clause

distlib	Python license
DLLogger	Apache2
dm-tree	Apache 2.0
docker-pycreds	Apache License 2.0
docopt	MIT
docutils	public domain Python 2-Clause BSD GPL 3
dunamai	MIT
einops	Apache License 2.0
entrypoints	MIT License
environment-kernels	BSD
exceptiongroup	MIT
executing	MIT
expecttest	MIT
faerun	MIT License
fairseq	MIT License
fastapi	MIT License
fastapi-utils	MIT
fasteners	ASL 2.0
fastjsonschema	BSD
fastparquet	Apache License 2.0
filelock	Public Domain (Unlicense)
flake8	MIT
flash-attn	Apache License 2.0

flask	BSD 3-Clause License
flatbuffers	Apache 2.0
flit-core	BSD 3-Clause License
fonttools	MIT
fqdn	MPL 2.0

freetype-py	BSD 2-Clause License
frozenlist	Apache 2
fsspec	BSD
future	MIT
fuzzywuzzy	GPLv2
gast	BSD 3-Clause
gcovr	BSD
gcsfs	BSD
gensim	LGPL-2.1-only
geometric	MIT License
gitdb	BSD License
GitPython	BSD
gmpy2	LGPL-3.0+
google-api-core	Apache 2.0
googleapis-common-protos	Apache-2.0
google-auth	Apache 2.0
google-auth-oauthlib	Apache 2.0
google-cloud-core	Apache 2.0
google-cloud-storage	Apache 2.0
google-crc32c	Apache 2.0
google-pasta	Apache 2.0
google-resumable-media	Apache 2.0
gpustat	MIT
graphormer-pretrained	MIT
greenlet	MIT License
GridDataFormats	LGPLv3
grpcio	Apache License 2.0
gsd	BSD - 2 clause

h11	MIT
h2	MIT License

h5py	BSD
hjson	MIT License
hpack	MIT License
httpcore	BSD 3-Clause License
httpx	BSD 3-Clause License
huey	BSD 3-Clause License
huggingface-hub	Apache
hydra-core	MIT
hyperframe	MIT License
idna	BSD 3-Clause License
igraph	GNU General Public License (GPL)
imagecodecs-lite	BSD
imageio	BSD-2-Clause
imagesize	MIT
importlib-metadata	Apache License 2.0
importlib-resources	Apache License 2.0
inflect	MIT License
iniconfig	MIT License
ipykernel	BSD 3-Clause License
ipyplot	MIT
ipython	BSD
ipython-genutils	BSD
ipywidgets	BSD 3-Clause License
isodate	BSD
isoduration	Apache License 2.0
isort	MIT

iteration-utilities	Apache License Version 2.0
itsdangerous	BSD-3-Clause
jaraco.collections	MIT License
jaraco.context	MIT License
jaraco.functools	MIT License
jaraco.text	MIT License
jarowinkler	MIT
jedi	MIT
Jinja2	BSD-3-Clause

jinja2-ansible-filters	GPL3
jmespath	MIT
joblib	BSD
json5	Apache
jsonpointer	Modified BSD License
jsonschema	MIT
jsonschema-specifications	MIT
jupyter	BSD
jupyter-client	BSD 3-Clause License
jupyter-console	BSD 3-Clause License
jupyter-contrib-core	BSD 3-clause
jupyter-contrib-nbextensions	BSD
jupyter-core	BSD 3-Clause License
jupyter-events	BSD 3-Clause License
jupyter-highlight-selectedword	BSD
jupyterlab	Copyright (c) 2015-2022 Project Jupyter Contributors
jupyterlab-pygments	Copyright (c) 2015 Project Jupyter Contributors

jupyterlab-server	Copyright (c) 2015-2017 Project Jupyter Contributors
jupyterlab-widgets	BSD-3-Clause
jupyter-latex-envs	Modified BSD
jupyter-lsp	BSD-3-Clause
jupyter-nbextensionsconfigurator	BSD 3-clause
jupyter-server	BSD 3-Clause License
jupyter-server-terminals	BSD 3-Clause License
keras	Apache 2.0
Keras-Applications	MIT
Keras-Preprocessing	MIT
kiwisolver	BSD 3-Clause License
lark	MIT
lark-parser	MIT
lazy-object-proxy	BSD-2-Clause
Levenshtein	GPL
libclang	Apache License 2.0

lightgbm	The MIT License (Microsoft)
lightning-utilities	Apache-2.0
littleutils	MIT
llvmlite	BSD
lmdb	OLDAP-2.8
locket	BSD-2-Clause
loguru	MIT license
lxml	BSD
Mako	MIT
MAP4	MIT License
Markdown	BSD License

MarkupSafe	BSD-3-Clause
matplotlib	PSF
matplotlib-inline	BSD 3-Clause
matplotlib-venn	MIT
mccabe	Expat license
MDAnalysis	GNU General Public License v2 (GPL-2.0)
mdtraj	LGPLv2.1+
meeko	LGPL-2.1
mhfp	MIT License
mistune	BSD-3-Clause
mkl-fft	BSD
mkl-random	BSD
mkl-service	BSD
ml-collections	Apache 2.0
mmcif-pdbx	MIT License
mmtf-python	Apache 2.0
mock	BSD 3-Clause License
mol2vec	BSD 3-clause
molfeat	Apache
mols2grid	Apache License Version 2.0
mordred	BSD-3-Clause
more-itertools	MIT License
mpmath	BSD
mrcfile	BSD

msgpack	Apache 2.0
multidict	Apache 2
multiprocess	BSD

multisplitby	MIT
munkres	Apache Software License
mypy-extensions	MIT License
mysqlclient	GPL
nbclient	BSD 3-Clause License
nb-conda-kernels	BSD 3-Clause License
nbconvert	BSD 3-Clause License
nbformat	BSD 3-Clause License
nest-asyncio	BSD
netCDF4	License :: OSI Approved :: MIT License
networkx	BSD 3-Clause License
ninja	Apache 2.0
notebook	BSD 3-Clause License
notebook-shim	BSD 3-Clause License
numba	BSD
numexpr	MIT
numpy	BSD
nvidia-ml-py	BSD
nvidia-pyindex	Apache2
oauthlib	BSD
ogb	MIT
olefile	BSD
omegaconf	BSD 3-Clause License
opencensus	Apache-2.0
opencensus-context	Apache-2.0
OpenEye-toolkits-python3linux-x64	Other/Proprietary License
openfold	Apache License Version 2.0
OpenMM	Python Software Foundation License (BSD-like)

opensearch-py	Apache-2.0
opt-einsum	MIT
OptKing	BSD-3C

optuna	MIT License
os-command-py	MIT License
outdated	MIT
overrides	Apache License Version 2.0
Package	License
packaging	Apache License 2.0 or BSD 2-Clause License
pandas	BSD-3-Clause
pandocfilters	BSD-3-Clause
paramiko	LGPL
ParmEd	LGPL
parso	MIT
partd	BSD
pathos	3-clause BSD
pathspec	MPL 2.0 License
pathtools	MIT License
patsy	2-clause BSD
pbr	Apache License 2.0
pdb2pqr	BSD
pdbfixer	MIT
pdb-manip-py	MIT License
pexpect	ISC license
pickleshare	MIT
Pillow	HPND
Pint	BSD
pip	MIT

pkgutil-resolve-name	MIT License
platformdirs	MIT License
pluggy	MIT
plumbum	Copyright (c) 2013 Tomer Filiba (tomerfiliba@gmail.com)
ply	BSD
pmapper	MIT License
Pmw	BSD
pooch	BSD 3-Clause License
portalocker	BSD-3-Clause
portend	MIT License

pox	3-clause BSD
ppft	BSD-like
prettytable	BSD (3 clause)
ProDy	MIT License
prolif	Apache License Version 2.0
prometheus-client	Apache Software License 2.0
promise	MIT
prompt-toolkit	BSD 3-Clause License
propka	LGPL v2.1
propy3	GPLv2
protobuf	3-Clause BSD License
psutil	BSD-3-Clause
psycopg2	LGPL with exceptions
psycopg2-binary	LGPL-3.0 License
ptyprocess	ISC License
pure-eval	MIT
py	MIT license

py3Dmol	MIT
pyarrow	Apache License Version 2.0
pyasn1	BSD
pyasn1-modules	BSD-2-Clause
pybel	Apache 2.0 License
pycairo	LGPL-2.1 License
pycodestyle	Expat license
pycparser	BSD
py-cpuinfo	MIT
pycrypto	Public Domain
pydantic	MIT
pydantic-core	MIT
pyDeprecate	MIT
pyedr	LGPLv2+
pyflakes	MIT
Pygments	BSD License
pygraphviz	BSD
PyJWT	MIT

pylint	GPL-2.0-or-later
pymol	BSD 2-Clause License
PyNaCl	Apache License 2.0
pynndescent	BSD
pyOpenSSL	Apache License Version 2.0
pyparsing	MIT License
PyQt5	GPL v3
PyQt5-sip	SIP
pyrsistent	MIT

pysftp	BSD
pyslurm	GPLv2
PySocks	BSD
py-spy	MIT
PyTDC	MIT
pytest	MIT
pytest-cov	MIT
pytest-mock	MIT
python-dateutil	Dual License
python-dotenv	BSD-3-Clause
python-igraph	GNU General Public License (GPL)
python-json-logger	BSD
python-Levenshtein	GPL-2.0-or-later
pytng	GNU Lesser General Public License v2.1 (LGPL-2.1)
pytorch-lightning	Apache-2.0
pytz	MIT
pyu2f	Apache 2.0
PyWavelets	MIT
PyYAML	MIT
pyyaml-include	GPLv3+
pyzmq	LGPL+BSD
qcelemental	BSD-3-Clause
qcengine	BSD-3C
qtconsole	BSD
QtPy	MIT
questionary	MIT

rapidfuzz	MIT
ray	Apache 2.0

rdflib	BSD-3-Clause
rdkit	BSD-3-Clause
rdkit-pypi	BSD-3-Clause
redis	MIT
referencing	MIT
regex	Apache Software License
reinvent-chemistry	MIT License
reinvent-scoring	MIT License
reportlab	BSD license Copyright (c) 2000-2018 ReportLab Inc.
requests	Apache 2.0
requests-file	Apache 2.0
requests-oauthlib	ISC
resp	BSD license
rfc3339-validator	MIT license
rfc3986-validator	MIT license
rlPyCairo	BSD license Copyright (c) 2000-2022 ReportLab Inc.
rpds-py	MIT
rsa	Apache-2.0
s3fs	BSD
s3transfer	Apache License 2.0
sacrebleu	Apache License 2.0
safetensors	Apache License 2.0
scikit-image	Modified BSD
scikit-learn	new BSD
scipy	BSD
seaborn	BSD 3-Clause License
selfies	MIT License
Send2Trash	BSD License
sentry-sdk	BSD

setproctitle	BSD
setuptools	MIT License
shortuuid	BSD
sip	SIP

six	MIT
sklearn	BSD 3-Clause License
smart-open	MIT
smmap	BSD
sniffio	MIT OR Apache-2.0
snowballstemmer	BSD-3-Clause
soupsieve	MIT License
sphinx	BSD 2-Clause License
sphinxcontrib-applehelp	BSD-2-Clause
sphinxcontrib-devhelp	BSD
sphinxcontrib-htmlhelp	BSD-2-Clause
sphinxcontrib-jsmath	BSD
sphinxcontrib-qthelp	BSD
sphinxcontrib-serializinghtml	BSD
sphinx-rtd-theme	MIT
SQLAlchemy	MIT
stack-data	MIT
starlette	BSD 3-Clause License
statsmodels	BSD License
svgutils	BSD 3-Clause License
sympy	BSD
tables	BSD 3-Clause License
tabulate	MIT

ta-ml	MIT License
ta-screen	TBA
tempora	MIT License
tensorboard	Apache 2.0
tensorboard-data-server	Apache 2.0
tensorboard-plugin-wit	Apache 2.0
tensorboardX	MIT license
tensorflow	Apache 2.0
tensorflow-estimator	Apache 2.0
tensorflow-io-gcs-filesystem	Apache License 2.0
termcolor	MIT
terminado	BSD 2-Clause License

texttable	MIT
threadpoolctl	BSD-3-Clause
tidynamics	BSD
tifffile	BSD
tinycss2	BSD 3-Clause License
tmap	BSD 3-Clause License
tokenizers	Apache License 2.0
toml	MIT
tomli	MIT License
tomlkit	MIT License
toolz	BSD
torch	BSD-3
torchani	MIT
torchaudio	BSD 3-Clause License
torch-geometric	MIT License

torchmetrics	Apache-2.0
torchvision	BSD
tornado	http://www.apache.org/licenses/LICENSE-2.0
tqdm	MPLv2.0 MIT Licences
traitlets	BSD 3-Clause License
transformers	Apache 2.0 License
trimesh	The MIT License (MIT)
triton	Apache License 2.0
typed-ast	Apache License 2.0
types-python-dateutil	Apache-2.0 license
typing-extensions	Python Software Foundation License
typing-utils	Apache License 2.0
tzdata	Apache-2.0
ujson	BSD 3-Clause License
unicodedata2	Apache License 2.0
uri-template	MIT License
urllib3	MIT
uvicorn	BSD 3-Clause License
versioneer	This is free and unencumbered software released into the public domain.
vina	Apache-2.0
virtualenv	MIT
wandb	MIT license
wcwidth	MIT
webcolors	BSD-3-Clause
webencodings	BSD
websocket-client	Apache-2.0
Werkzeug	BSD-3-Clause
wheel	MIT

widgetsnbextension	BSD 3-Clause License
wrapt	BSD
xgboost	Apache-2.0
xlrd	BSD
xlutils	MIT
xlwt	BSD
xmltodict	MIT
xtb	LGPL-3.0-or-later
xyzservices	3-Clause BSD
yarl	Apache-2.0
zc.lockfile	ZPL 2.1
zipp	MIT License
zope.deprecation	ZPL 2.1