

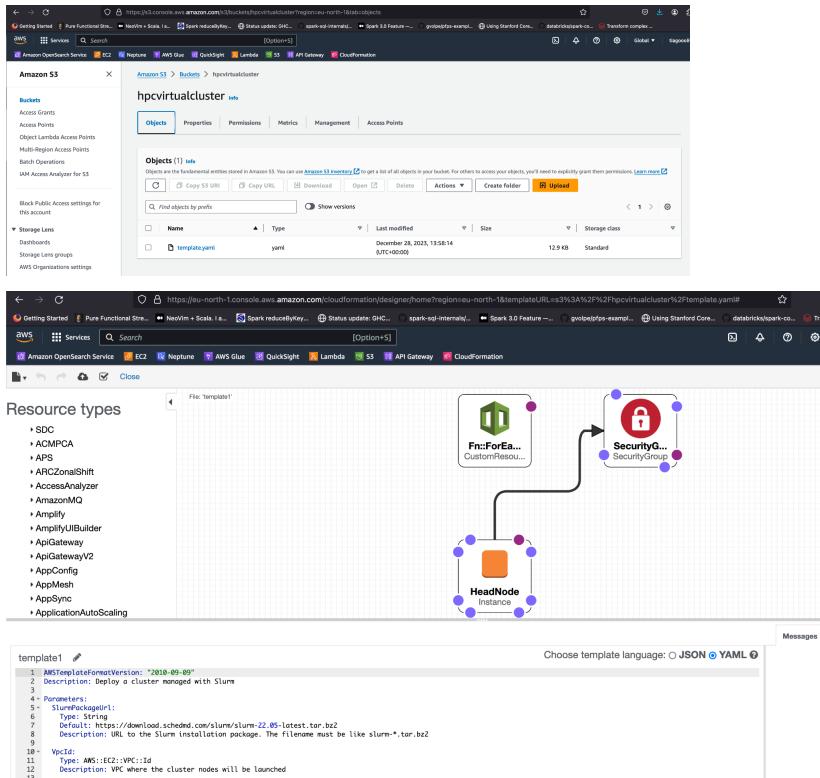
HPC - Cloud - Virtual Cluster

Student: Tiago de Souza Oliveira

To create the AWS Virtual Cluster HPC environment

The design on the Infrastructure as Code (.yml) for AWS:CloudFormation leveraging the code from <https://github.com/micap-hpcn/virtual-clusters/blob/main/template.yaml>

From AWS:S3, with the caveat that the region for S3 and CloudFormation should match.



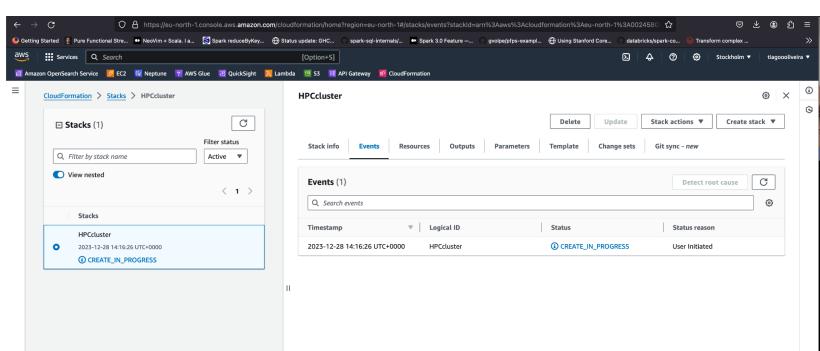
The screenshot shows the AWS CloudFormation Designer interface. At the top, there's a navigation bar with tabs like 'Getting Started', 'Pure Functional Str...', 'NewVM + Scale I...', 'Spark reduced...', 'Status update: SH...', 'spark-sql-inter...', 'Spark 3.0 Feature...', 'gotopclips-exam...', 'Using Stanford Cor...', 'databrickspark-co...', 'Transform complex...', 'AWS Lambda', 'QuickSight', 'Lambda', 'S3', 'API Gateway', and 'CloudFormation'. Below the navigation, there's a search bar and a dropdown menu for 'Option+5'. The main area has sections for 'Resource types' (listing various AWS services like S3, ACMPCA, APS, etc.) and 'Components' (with a 'Template' tab selected). A preview window shows a network diagram with nodes: 'Fn::FindInMap' (CustomResource), 'SecurityGroup' (SecurityGroup), and 'HeadNode' (Instance). Below the preview is a code editor for 'template1.yaml' containing the CloudFormation template code. At the bottom right, there's a 'Choose template language' dropdown set to 'YAML'.

```
template1.yaml
1: #!/bin/bash
2: #!/bin/bash
3: #!/bin/bash
4: #!/bin/bash
5: #!/bin/bash
6: #!/bin/bash
7: #!/bin/bash
8: #!/bin/bash
9: #!/bin/bash
10: #!/bin/bash
11: #!/bin/bash
12: #!/bin/bash
13: #!/bin/bash
14: #!/bin/bash
15: #!/bin/bash
16: #!/bin/bash
17: #!/bin/bash
18: #!/bin/bash
```

template1.yaml

```
1: AWSTemplateFormatVersion: '2010-09-09'
2: Description: Deploy a cluster managed with Slurm
3:
4: Parameters:
5:   SlurmPkgUrl:
6:     Type: String
7:     Default: https://download.schedmd.com/slurm/slurm-12.05-latest.tar.bz2
8:     Description: URL to the Slurm installation package. The filename must be like slurm-*.tar.bz2
9:
10: VpcId:
11:   Type: AWS::EC2::VPC
12:   Description: VPC where the cluster nodes will be launched
13:
14: HeadNodeAZ:
15:   Type: AWS::EC2::AvailabilityZone
16:   Description: Availability Zone where the head node will be launched
17:
18: Kernels:
```

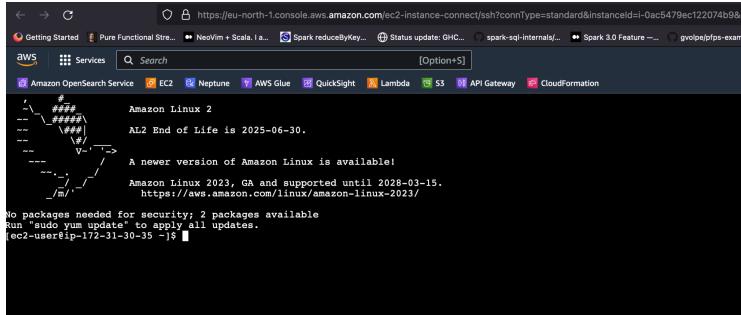
Components [Template](#)



The screenshot shows the AWS CloudFormation Stacks interface. At the top, there's a navigation bar with tabs like 'Getting Started', 'Pure Functional Str...', 'NewVM + Scale I...', 'Spark reduced...', 'Status update: SH...', 'spark-sql-inter...', 'Spark 3.0 Feature...', 'gotopclips-exam...', 'Using Stanford Cor...', 'databrickspark-co...', 'Transform complex...', 'AWS Lambda', 'QuickSight', 'Lambda', 'S3', 'API Gateway', and 'CloudFormation'. Below the navigation, there's a search bar and a dropdown menu for 'Option+5'. The main area shows a list of stacks under 'Stacks (1)'. One stack is listed: 'HPCCluster' (Status: CREATE_IN_PROGRESS, Last updated: 2023-12-28 14:16:26 UTC+0000). Below the stack list is a table for 'Events (1)' showing a single event: 'Timestamp: 2023-12-28 14:16:26 UTC+0000 | Logical ID: HPCCluster | Status: CREATE_IN_PROGRESS | Status reason: User Initiated'.

Since the cloudformation was complaining about a missing role, I had to create a new one with full privileges on EC2. Then it worked as expected.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 address	Private IP add.	Monitoring
HPCVirtualCluster-headnode	i-0ac5479ec122074b980	Running	c5.large	2/2 checks passed	No alarms	eu-north-1a	13.51.109.127	172.31.30.35	disabled
HPCVirtualCluster-on2	i-0d5e5ab0125f527	Running	c5.large	2/2 checks passed	No alarms	eu-north-1a	51.20.107.131	172.31.25.135	disabled
HPCVirtualCluster-on3	i-09550116b059445	Running	c5.large	2/2 checks passed	No alarms	eu-north-1a	16.16.185.55	172.31.21.115	disabled
HPCVirtualCluster-on1	i-00933220be0c145	Running	c5.large	2/2 checks passed	No alarms	eu-north-1a	16.171.110.176	172.31.20.132	disabled



```
'\      #
# \    #####
# \  \####\ Amazon Linux 2
# \  \#\#| AL2 End of Life is 2025-06-30.
# \  \#/ \-->
# \  V- \--> A newer version of Amazon Linux is available!
# \  / \-->
# \ / \--> Amazon Linux 2023, GA and supported until 2028-03-15.
# \ / \--> https://aws.amazon.com/linux/amazon-linux-2023/
/m/ \-->

No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-30-35 ~]$
```

```
'\      #
# \    #####
# \  \####\ Amazon Linux 2
# \  \#\#| AL2 End of Life is 2025-06-30.
# \  \#/ \-->
# \  V- \--> A newer version of Amazon Linux is available!
# \  / \-->
# \ / \--> Amazon Linux 2023, GA and supported until 2028-03-15.
# \ / \--> https://aws.amazon.com/linux/amazon-linux-2023/
/m/ \-->

No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-30-35 ~]$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
aws*   up infinite 3 idle cn[0-2]
[ec2-user@ip-172-31-30-35 ~]$ squeue
JOBID PARTITION NAME USER ST      TIME NODES NODELIST(REASON)
[ec2-user@ip-172-31-30-35 ~]$
```



```
ec2-user@ip-172-31-30-35 ~$ salloc -N 2
salloc: Granted job allocation 1
[ec2-user@ip-172-31-30-35 ~]$ scontrol show nodes
NodeName=cn0 Arch=x86_64 CoresPerSocket=1
CPUAlloc=2 CPUEffctv=2 CPUTot=2 CPULoad=0.01
AvailableFeatures=(null)
ActiveFeatures=(null)
Gres=(null)
NodeAddr=172.31.21.113 NodeHostName=ip-172-31-21-113 Version=22.05.11
OS=Linux 4.14.330-250.540.amzn2.x86_64 #1 SMP Tue Nov 21 09:57:48 UTC 2023
RealMemory=3719 AllocMem=0 FreeMem=2709 Sockets=1 Boards=1
State=ALLOCATED+DYNAMIC_NORM ThreadsPerCore=2 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
Partitions=aws
BootTime=2023-12-28T14:55:50 SlurmStart=2023-12-28T14:56:59
LastBusyTime=2023-12-28T14:56:59
CfgTRES=cpu=2,mem=3719M,billing=2
AllocTRES=cpu=2
CapWatts=n/a
CurrentWatts=0 AveWatts=0
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s

NodeName=cn1 Arch=x86_64 CoresPerSocket=1
CPUAlloc=2 CPUEffctv=2 CPUTot=2 CPULoad=0.00
AvailableFeatures=(null)
ActiveFeatures=(null)
Gres=(null)
NodeAddr=172.31.20.132 NodeHostName=ip-172-31-20-132 Version=22.05.11
OS=Linux 4.14.330-250.540.amzn2.x86_64 #1 SMP Tue Nov 21 09:57:48 UTC 2023
RealMemory=3719 AllocMem=0 FreeMem=2771 Sockets=1 Boards=1
State=ALLOCATED+DYNAMIC_NORM ThreadsPerCore=2 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
Partitions=aws
BootTime=2023-12-28T14:55:50 SlurmStart=2023-12-28T14:56:59
LastBusyTime=2023-12-28T14:56:59
CfgTRES=cpu=2,mem=3719M,billing=2
AllocTRES=cpu=2
CapWatts=n/a
CurrentWatts=0 AveWatts=0
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s

NodeName=cn2 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUEffctv=2 CPUTot=2 CPULoad=0.00
AvailableFeatures=(null)
ActiveFeatures=(null)
Gres=(null)
NodeAddr=172.31.25.135 NodeHostName=ip-172-31-25-135 Version=22.05.11
OS=Linux 4.14.330-250.540.amzn2.x86_64 #1 SMP Tue Nov 21 09:57:48 UTC 2023
RealMemory=3719 AllocMem=0 FreeMem=2771 Sockets=1 Boards=1
State=IDLE+DYNAMIC_NORM ThreadsPerCore=2 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
Partitions=aws
BootTime=2023-12-28T14:55:50 SlurmStart=2023-12-28T14:57:03
LastBusyTime=2023-12-28T14:57:03
CfgTRES=cpu=2,mem=3719M,billing=2
AllocTRES=
CapWatts=n/a
CurrentWatts=0 AveWatts=0
```

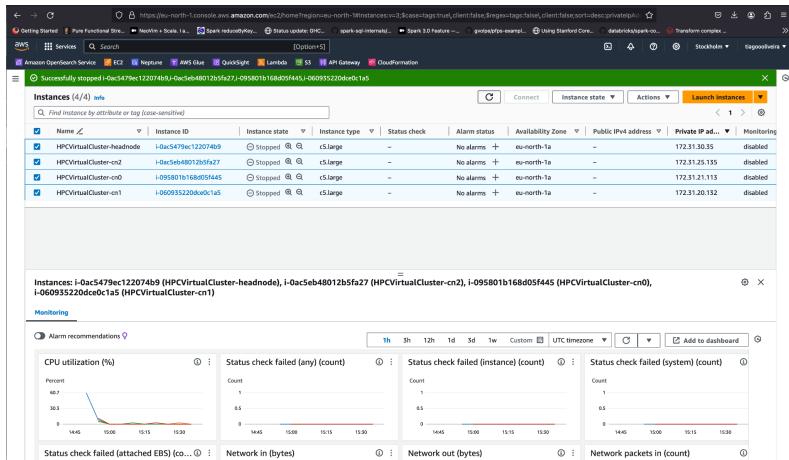
```

MPI_Comm_size(MPI_COMM_WORLD, &world_size);
// Get the rank of the process
int world_rank;
MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
// Get the name of the processor
char processor_name[MPI_MAX_PROCESSOR_NAME];
int name_len;
MPI_Get_processor_name(processor_name, &name_len);
// Print off a hello world message
printf("Hello world from processor %s, rank %d out of %d processors\n",
processor_name, world_rank, world_size);
// Finalize the MPI environment.
MPI_Finalize();
}

[ec2-user@ip-172-31-30-35 mpi]$ mpicc -o hello hello.c
[ec2-user@ip-172-31-30-35 mpi]$ ./hello
Hello world from processor ip-172-31-30-35.eu-north-1.compute.internal, rank 0 out of 1 processors
[ec2-user@ip-172-31-30-35 mpi]$ nano hello.sh
[ec2-user@ip-172-31-30-35 mpi]$ ls
hello hello.c hello.sh
[ec2-user@ip-172-31-30-35 mpi]$ cat hello.sh
#!/bin/bash
#
#SBATCH --job-name=hello
#SBATCH --output=hello.out
#SBATCH --partition=aws
#
#SBATCH --time=10:00
#SBATCH --cpus-per-task=1

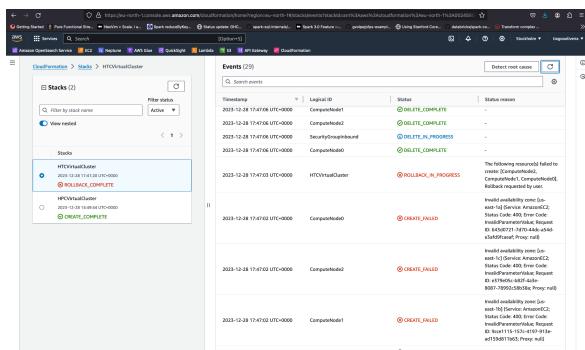
mpirun hello
[ec2-user@ip-172-31-30-35 mpi]$ 
[ec2-user@ip-172-31-30-35 mpi]$ 
[ec2-user@ip-172-31-30-35 mpi]$ 
[ec2-user@ip-172-31-30-35 mpi]$ 
[ec2-user@ip-172-31-30-35 mpi]$ sbatch -n 2 hello.sh
Submitted batch job 3
[ec2-user@ip-172-31-30-35 mpi]$ squeue
      JOBID PARTITION   NAME   USER ST       TIME NODELIST(REASON)
[ec2-user@ip-172-31-30-35 mpi]$ cat hello.out
Hello world from processor ip-172-31-21-113.eu-north-1.compute.internal, rank 0 out of 2 processors
Hello world from processor ip-172-31-21-113.eu-north-1.compute.internal, rank 1 out of 2 processors
[ec2-user@ip-172-31-30-35 mpi]$ sbatch -N 2 hello.sh
Submitted batch job 3
[ec2-user@ip-172-31-30-35 mpi]$ cat hello.out
Hello world from processor ip-172-31-21-113.eu-north-1.compute.internal, rank 0 out of 4 processors
Hello world from processor ip-172-31-20-132.eu-north-1.compute.internal, rank 2 out of 4 processors
Hello world from processor ip-172-31-21-113.eu-north-1.compute.internal, rank 1 out of 4 processors
Hello world from processor ip-172-31-20-132.eu-north-1.compute.internal, rank 3 out of 4 processors
[ec2-user@ip-172-31-30-35 mpi]$ 

```



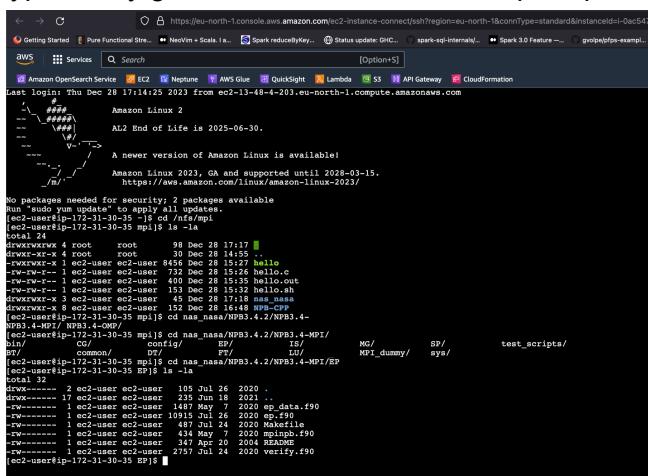
[WiP] To create the AWS Virtual Cluster HTC environment

For the HTC cluster option, the provisioning did not work well. It says the availability zone is invalid. So I have to choose a different one to test.



A Kernel for Intensive in Computation: Embarrassingly Parallel

Files from www.nas.nasa.gov/assets/npb/NPB3.4.2.tar.gz regarding EP kernel to compile and run to evaluate intensive computation in a AWS Virtual Cluster HPC having different EC2 type/family/generation instances to compare performance.



```
Last login: Thu Dec 28 17:14:25 2023 from ec2-13-46-4-203.eu-north-1.compute.amazonaws.com
[ec2-user@ip-172-31-30-35 ~]$ ls -l
total 24
drwxr-xr-x 4 root   root    98 Dec 28 17:17 .
drwxr-xr-x 1 ec2-user ec2-user 38 Dec 28 14:59 ..
-rw-rw-r-- 1 ec2-user ec2-user  1 Dec 28 14:59 hello
-rw-rw-r-- 1 ec2-user ec2-user 732 Dec 28 15:26 hello.out
-rw-rw-r-- 1 ec2-user ec2-user 400 Dec 28 15:35 hello.out~
-rw-rw-r-- 1 ec2-user ec2-user 15 Dec 28 17:18 nasa_nasa
drwxr-xr-x 3 ec2-user ec2-user 15 Dec 28 17:18 nasa_nasa
drwxr-xr-x 8 ec2-user ec2-user 152 Dec 28 16:48 NPB-CPP
[ec2-user@ip-172-31-30-35 ~]$ cd nasa_nasa/NPB3.4.2/NPB3.4-
NPB3.4-MPI/ NPB3.4-MPI/
[ec2-user@ip-172-31-30-35 ~]$ cd nasa_nasa/NPB3.4.2/NPB3.4-MPI/EP
[ec2-user@ip-172-31-30-35 ~]$ ls -l
total 32
drwxr-xr-x 2 ec2-user ec2-user 109 Jul 26 2020 .
drwxr-xr-x 17 ec2-user ec2-user 239 Jun 18 2021 ..
-rw-rw-r-- 1 ec2-user ec2-user 1487 May 7 2020 ep_data.f90
-rw-rw-r-- 1 ec2-user ec2-user 1048576 Jul 26 2020 ep_data.f90~
-rw-rw-r-- 1 ec2-user ec2-user 487 Jul 24 2020 Makefile
-rw-rw-r-- 1 ec2-user ec2-user 434 May 7 2020 mpinbg.f90
-rw-rw-r-- 1 ec2-user ec2-user 2754 Jul 26 2020 verify.f90
-rw-rw-r-- 1 ec2-user ec2-user 2757 Jul 24 2020 verify.f90
[ec2-user@ip-172-31-30-35 ~]$
```

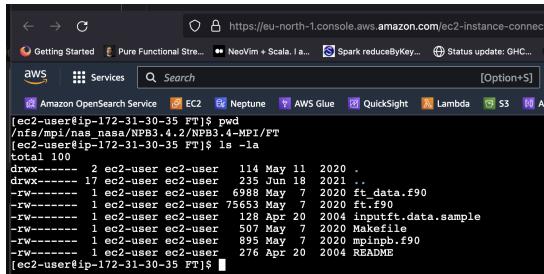
make ep CLASS=A

mpirun --use-hwthread-cpus -n 3 ./NPB3.4.2/NPB3.4-MPI/bin/ep.D.x

	Type: ; Family: ; Gen:	Type: ; Family: ; Gen:	Type: ; Family: ; Gen:
Execution	Time	Time	Time
#1			
#2			
#3			
#4			
#5			
AVG			

A Kernel for Intensive in Communication: 3D fast Fourier Transform

Files from www.nas.nasa.gov/assets/npb/NPB3.4.2.tar.gz regarding FT kernel to compile and run to evaluate intensive communication in a AWS Virtual Cluster HPC and HTC.



```
[ec2-user@ip-172-31-30-35 FT]$ pwd  
/nfs/mpi/nas_nasa/NPB3.4.2/NPB3.4-MPI/FT  
[ec2-user@ip-172-31-30-35 FT]$ ls -la  
total 100  
drwxr-xr-x 2 ec2-user ec2-user 114 May 11 2020 .  
drwxr-xr-x 17 ec2-user ec2-user 235 Jun 18 2021 ..  
-rw-r--r-- 1 ec2-user ec2-user 6988 May 7 2020 ft.f90  
-rw-r--r-- 1 ec2-user ec2-user 75653 May 7 2020 ft.f90  
-rw-r--r-- 1 ec2-user ec2-user 128 Apr 20 2004 inputft.data.sample  
-rw-r--r-- 1 ec2-user ec2-user 507 May 7 2020 Makefile  
-rw-r--r-- 1 ec2-user ec2-user 895 May 7 2020 mpinpnb.f90  
-rw-r--r-- 1 ec2-user ec2-user 276 Apr 20 2004 README  
[ec2-user@ip-172-31-30-35 FT]$
```

make ft CLASS=A

mpirun --use-hwthread-cpus -n 3 ./NPB3.4.2/NPB3.4-MPI/bin/ft.D.x

	Type: ; Family: ; Gen:	Type: ; Family: ; Gen:	Type: ; Family: ; Gen:
Execution	Time	Time	Time
#1			
#2			
#3			
#4			
#5			
AVG			

References

<https://www.nas.nasa.gov/assets/npb/NPB3.4.2.tar.gz>
<http://www.nas.nasa.gov/Software/NPB/>
<https://microsoft.github.io/VirtualClient/docs/workloads/nasparallel/>
git clone <https://github.com/GMAP/NPB-CPP.git>
<https://stackoverflow.com/questions/71222043/nas-parallel-benchmarks-mpi-segmentation-fault-executing-ft-and-mg-for-class-d>
<https://github.com/cwang9208/NPB>