

Check 1+1 MKSP Redundancy Status

May 2020

Introduction

When using 1+1 Stream Processors (and for this example I am using 1+1 SP in Compact Mode). The 1+1 SP servers

There are several operations that the SPs are handling, and either SP can be the one doing an operation, these are:

```
StatMux
PSI
Scrambling CA
MUX
```

I am unsure of how tightly coupled these components are, i.e. do Mux and PSI always follow each other, I am unsure. StatMux and CA do seem to be quite independent though.

Please note - You must use the **Management VIP** for the IP Address for this to work.

In a web browser, enter the following:

```
http://your_MKSP_VIP_address:8080/api/mfsp/activeStandby/status?version=11.3
```

You will see an output in your browser similar to below (with different instance names) - **If your output is one long line or lines of text, then install a JSON viewer plugin in to your browser.**

```
{
  mongo_status: "ok",
  rabbit_status: "ok",
  statmux: {
    active: {
      SP01: [
        "MK01"
      ],
      SP02: [ ]
    },
    inactive: {
      SP01: [ ],
      SP02: [
        "MK01"
      ]
    }
  },
  psi: {
    active: {
      SP01: [
        "mux.mux_52_1_1.outputs.TS1"
      ],
      SP02: [ ]
    },
    inactive: {
      SP01: [ ],
      SP02: [
        "mux.mux_52_1_1.outputs.TS1"
      ]
    }
  },
  mux: {
    active: {
      SP01: [
        "mux_52_1_1"
      ],
      SP02: [ ]
    }
  }
}
```

```

    },
    inactive: {
      SP01: [ ],
      SP02: [
        "mux_52_1_1"
      ]
    }
  },
  simulcrypt: {
    active: {
      SP01: [
        "mux_52_1_1"
      ],
      SP02: [ ]
    },
    inactive: {
      SP01: [ ],
      SP02: [
        "mux_52_1_1"
      ]
    }
  }
}
}

```

In the output above we can see that **Mongo** and **Rabbit** status are **ok**. **statmux**, **psi**, **mux** and **simulcrypt** are all **active** on **SP01** (this is the active SP on the nCompass ESM).

Now I will do a Mux switch from nCompass ESM.

```

{
  mongo_status: "ok",
  rabbit_status: "ok",
  mux: {
    active: {
      SP01: [ ],
      SP02: [
        "mux_52_1_1"
      ]
    },
    inactive: {
      SP01: [
        "mux_52_1_1"
      ],
      SP02: [ ]
    }
  },
  psi: {
    active: {
      SP01: [ ],
      SP02: [
        "mux.mux_52_1_1.outputs.TS1"
      ]
    },
    inactive: {
      SP01: [
        "mux.mux_52_1_1.outputs.TS1"
      ],
      SP02: [ ]
    }
  },
  statmux: {
    active: {
      SP01: [ ],
      SP02: [
        "MK01"
      ]
    },
    inactive: {
      SP01: [

```

```
        "MK01"
      ],
      SP02: [ ]
    }
  },
  simulcrypt: {
    active: {
      SP01: [ ],
      SP02: [
        "mux_52_1_1"
      ]
    },
    inactive: {
      SP01: [
        "mux_52_1_1"
      ],
      SP02: [ ]
    }
  }
}
```

As we can see from above, all the mux components are now being handled by Mux2 (SPO2).

I have seen examples where some elements are being handled by one mux, and the remainder by the other mux. While it seems slightly odd that this happens, it is perfectly normal operation.

From:
<http://cameraangle.co.uk/> - WalkerWiki - wiki.alanwalker.uk

Permanent link:
http://cameraangle.co.uk/doku.php?id=check_1_1_mksp_redundancy_status

Last update: **2023/03/09 22:35**

